Article

Systematic Selective Limits Application Using Decision Making Engines to Enhance Safety in Highly Automated Vehicles

Divya Garikapati [*] , Yiting Liu , Zhaoyuan Huo

Posted Date: 29 January 2024

doi: 10.20944/preprints202401.2034.v1

Keywords: automated vehicles; connected vehicles; V2I; multi-criteria decision making (MCDM); Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS); machine learning; safety limits; cloud-based; multiple safety profiles; operational design domain(ODD)

Article

Systematic Selective Limits Application Using Decision Making Engines to Enhance Safety in Highly Automated Vehicles

Divya Garikapati *, Dr.Yiting Liu and Zhaoyuan Huo

Woven by Toyota, 900 Arastradero Rd, Palo Alto, CA 94304

Abstract: Safety limits application has always been a traditional approach to ensure the safe operation of electro-mechanical systems within many industries including automated vehicles, robotics, aerospace, traditional automotive, railways, manufacturing, etc. In all of these applications, control and safety limits are usually hard-coded into the production firmware and are fixed throughout the entire product life cycle. Currently, due to the evolving needs of automated systems like automated vehicles and robots, this traditional approach does not address all the use cases and scenarios to ensure safe operation. Especially for data-driven machine learning applications that constantly evolve and learn over time, it is important to be able to adjust the safety limits application strategy to be more flexible and adaptable based on different Operational Design Domains (ODDs) and scenarios. Our ITSC conference paper [4] introduced the concept of a new safety limits application strategy called the Dynamic Control Limits Application (DCLA) strategy that supports the flexible application of diverse limits profiles based on the parameters involved within the dynamic scenario at different layers of the Autonomy software stack.This paper extends the DCLA strategy to derive the complete methodology for safety limits application based on ODD elements, scenario identification and scenario classification using Decision Making Engines. It leverages the layered architecture introduced in the ITSC conference paper to implement the Decision Making (DM) algorithms. Another important extension is the use of cloud infrastructure that is based on the Vehicle-to-Infrastructure (V2I) technology to store the scenarios and the limits mapping to serve as a ground truth and/or a backup mechanism in case of errors or failures associated with the main Decision Making (DM) Engine. There is also a focus on providing a more comprehensive list of scenarios and a custom built experimental dataset to cover the maximum ODD elements, and multiple tables of safety limits to be chosen from, which eventually helps in creating different safety limits application profiles. These distinct safety limits application profiles are based on the scenario parameters that are perceived by the system upon which the Decision Making algorithms are applied or trained. This systematic approach can be used within the industry for any of the future automated vehicles and systems until Level 5 Autonomy.

Keywords: automated vehicles; connected vehicles; V2I; multi-criteria decision making (MCDM); Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS); machine learning; safety limits; cloud-based; multiple safety profiles; operational design domain(ODD)

1. Introduction

Automated vehicle controllability in any scenario especially in emergency situations plays a key role in the safety system design. Public acceptance [20],[26],[27] is another important factor that improves with the capabilities of the safety systems. More and more intelligent safety mechanisms need to be part of the automated vehicles to mitigate this risk in the future and cover all types of scenarios in a proactive and distinguishable way. Operational Design Domain (ODD) [9],[12],[13] has become increasingly important for Level 2, 3 and 4 automated vehicles [5] because their performance is usually judged based on the in-scope elements for ODD. For example, if the current ODD is only to handle Urban driving upto 45 mph, then the capabilities of the system would be defined to cover all the scenarios that the vehicle encounters in this particular ODD. The system architecture would be designed with this short-term goal in mind along with the future potential of expansion in the scope. The sensor detection ranges [28],[29],[30] are estimated based on this definition to be able to handle the scenarios involving operation upto 45 mph like being able to detect traffic lights for example. Similarly, the safety case [31],[32],[15] would include safety goals, safe state mechanisms that cover the in-scope scenarios in detail. The safety case also has another aspect of handling out-of-scope elements, which involves critical thinking and understanding the operational limitations of the system. Operational limitations on the actuators have been studied before but how to make these thresholds more dynamic and flexible based on the ODD and scenario parameters as they evolve over time and as we move towards higher levels of autonomy has not been considered enough. There is a need for making this limits application strategy more intelligent and adaptable over time. The transition from traditional vehicle dynamic safety limit tables to environmental and scenario based safety limits generated online based on the environment and scenario parameters offers several benefits:

- **Adaptability to Real-Time Conditions:** Environmental-based safety limits are dynamically generated in real-time, considering the current environmental conditions. This adaptability allows for more accurate and context-sensitive responses, as opposed to static tables that may not account for the variability in real-world driving scenarios.
- **Increased Safety:** By considering factors such as weather, road conditions, traffic density, and pedestrian activity, these dynamic limits can enhance safety. They allow the vehicle to adjust its behavior in response to immediate surroundings, reducing the risk of accidents due to unforeseen environmental factors.
- **Efficiency and Performance:** Dynamic limits can optimize vehicle performance by adjusting to the environment, ensuring that the vehicle operates efficiently under varying conditions. This can lead to better fuel efficiency, reduced wear and tear, and an overall smoother driving experience.
- **Scalability and Learning:** As driving environments evolve, traditional tables can become outdated. Dynamic limits, however, can evolve with the environment, continually improving and adapting to new conditions.

Addition of Decision Making (DM) capabilties is needed to make this transition and adaptability possible. Decision Making (DM) Engines using Multi-Criteria Decision Making (MCDM), Decision Trees and Random Forest helped solve numerous problems and has evolved over time to be data-driven [2] as more and more data is made available. Decision Making (DM) Engines are the systems that are designed to facilitate the Decision Making (DM) algorithms or processes required. Decision Making (DM) algorithm is a step-by-step set of instructions or a computational procedure designed to solve a specific problem. Operational limitations of the system are being addressed in this paper with various examples as mentioned in the Section 4. The ODD definition helped to further define and decompose the system behavior that is needed to be developed. Scenarios that are in-scope have been derived from the detailed ODD elements as mentioned in section 5 of the AVSC best practices [12] which in-turn helped with defining the required capabilities and a set of system behaviors. Figure 1 illustrates the complete workflow of defining the system behaviors starting from the ODD definition until the application of controls and safety limits [1] within the actuators.

Leveraging a simple version of these DM algorithms in all aspects of safety but with proper supervision and safety principles [11] makes the new approach more powerful and valuable. One good example of such algorithm has been discussed in Section 5 of this paper. This algorithm takes in the required driving scenario parameters [14] and outputs the classification of safety limits to be chosen to achieve a certain type of trajectory customized for each scenario. Figure 2 below covers all the extensions from the DCLA strategy [4] that this paper addresses like incorporating the concepts of system definition process, ODD definitions along with using Decision Making (DM) Engine, providing the complete methodology for selecting the right classification of limits in each of the driving scenarios while also leveraging the cloud database network or the V2I [3] capability to reduce the on-board processing and storage needed for the components in-vehicle during nominal and failure scenarios of certain algorithms.

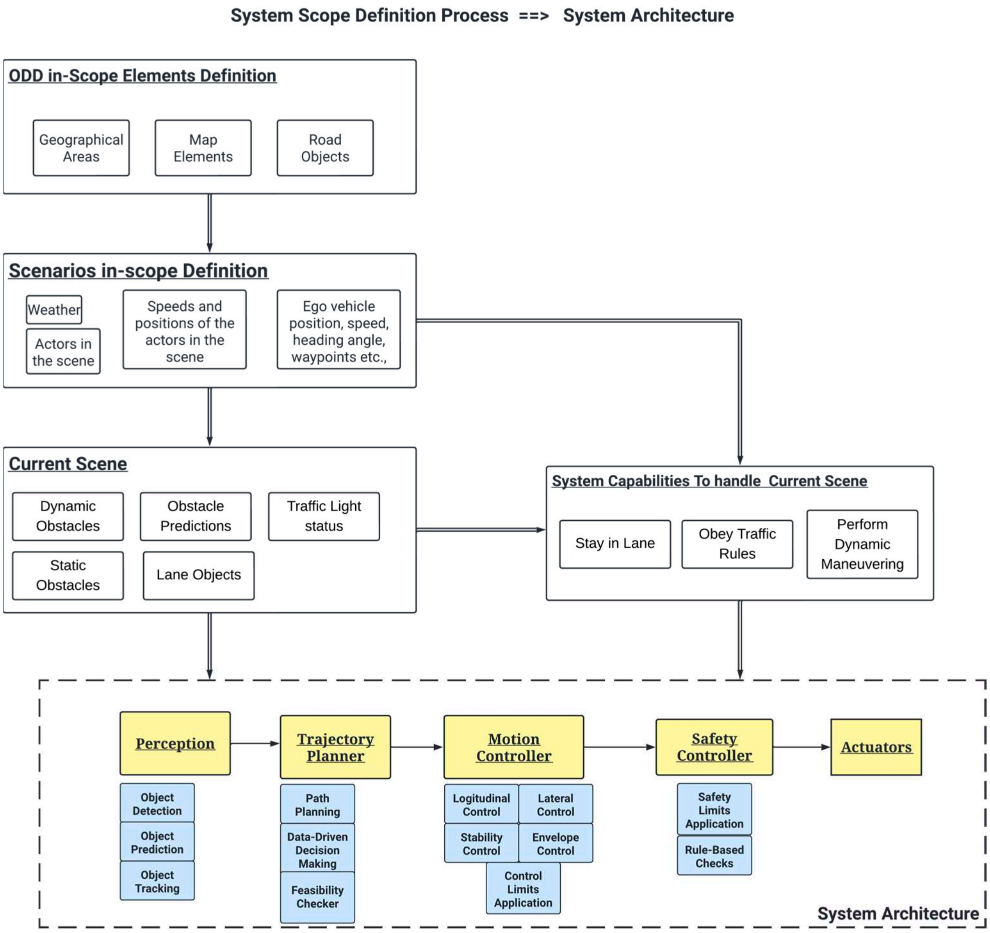

Figure 1. An Example of System Scope and Architecture Definition Processes.

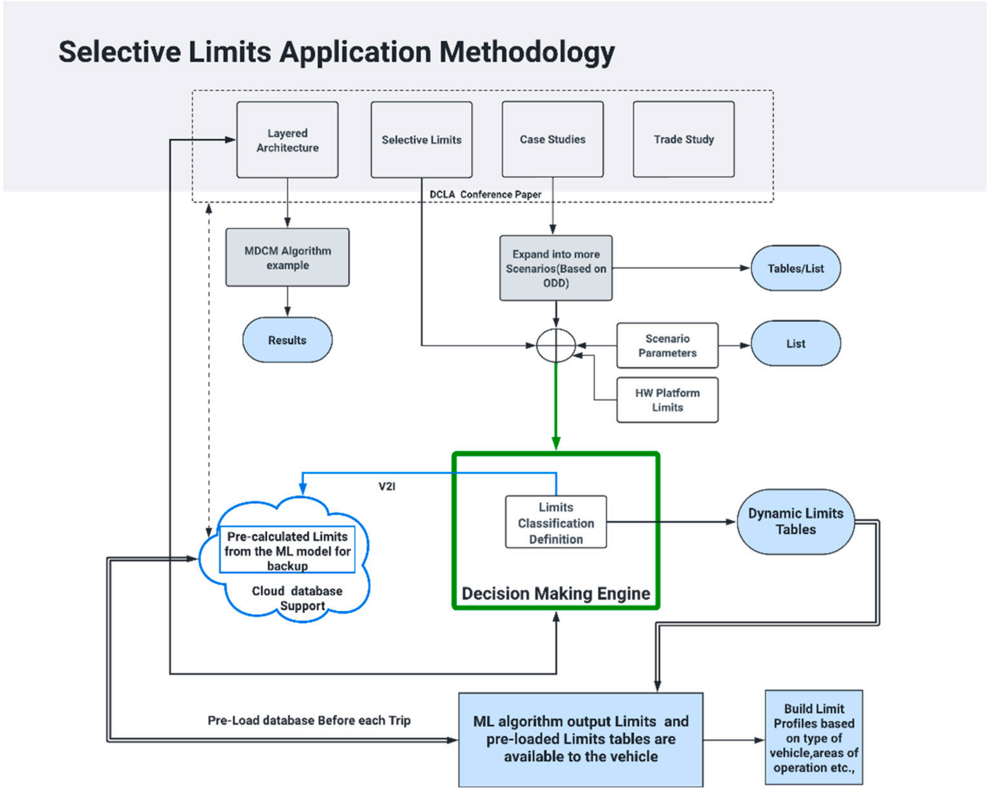
Selective Limits Application Methodology
Layered Architecture
Selective Limits
Case Studies
Trade Study
DCLA Conference Paper
MDCM Algorithm example
Results
Expand into more Scenarios(Based on ODD)
Tables/List
Scenario Parameters
List
HW Platform Limits
V2I
Pre-calculated Limits from the ML model for backup
Cloud database Support
Limits Classification Definition
Decision Making Engine
Dynamic Limits Tables
Pre-Load database Before each Trip
ML algorithm output Limits and pre-loaded Limits tables are available to the vehicle
Build Limit Profiles based on type of vehicle,areas of operation etc.,

Figure 2. Selective Limits Application Methodology : An Extension from the DCLA strategy framework.

2. Limits Evolution and Scenario Identification

Dynamic limits ensure safe vehicle maneuvering in automated driving. The safety controller sets constraints based on the vehicle state, such as current speed, and determines controlled commands or limits. If the actual controller command surpasses the determined limit, the safety controller's limiter logic is activated, adjusting the output value accordingly. This process, already in use in automated vehicles [33],[35], prevents exceeding safety thresholds, such as limiting steering angle values that could lead to disengagement if surpassed. In the case of applying steering angle rate limits, predefined limit values based on vehicle speed [43] are commonly hardcoded into the software, limiting flexibility across various scenarios. However, a more adaptive approach involves developing multiple tables dynamically selected based on the scenario, allowing greater customization during vehicle operation. A model predictive controller (MPC) [42] can then use these dynamic safety limit tables as constraints, ensuring controller commands align with established safety limits. The Decision Making (DM) Engine plays a crucial role in reliably detecting and classifying scenarios, applying the relevant set of limits. The DM algorithm dynamically adapts during automated vehicle operation, identifying and categorizing new scenarios. When a novel scenario is detected, it is sent to a cloud database using V2I [3], where it is mapped to an existing class of limits based on DM-extracted parameters and saved for future reference as shown in Figure 3. In the cases, where the newly identified scenario doesn't fit into existing classes, the DM algorithm signals this, and V2I-equipped vehicle systems notify the cloud database. Periodic checks by database maintainers ensure updates for new scenario classes. While our work currently focuses on four classes, this can adapt to evolving scenarios. This approach ensures the cloud database evolves with expanding Operational Design Domains (ODDs) and exploration of new aspects, reducing on-board storage needs, but comes with a caveat of the vehicles always needing to have continuous connectivity to the cloud infrastructure. To eliminate this shortcoming, the cloud database is only being used for storing all the mappings of "**Scenario Classes** <=> **Limits Tables**" that were already taken from the outputs of the DM algorithms in known scenarios previously as shown in Figure 3. Pre-loading the limits cloud database along with maps information within the vehicle before starting the trip is going to make this available to the system. This serves as a solution in case of network failures as well.

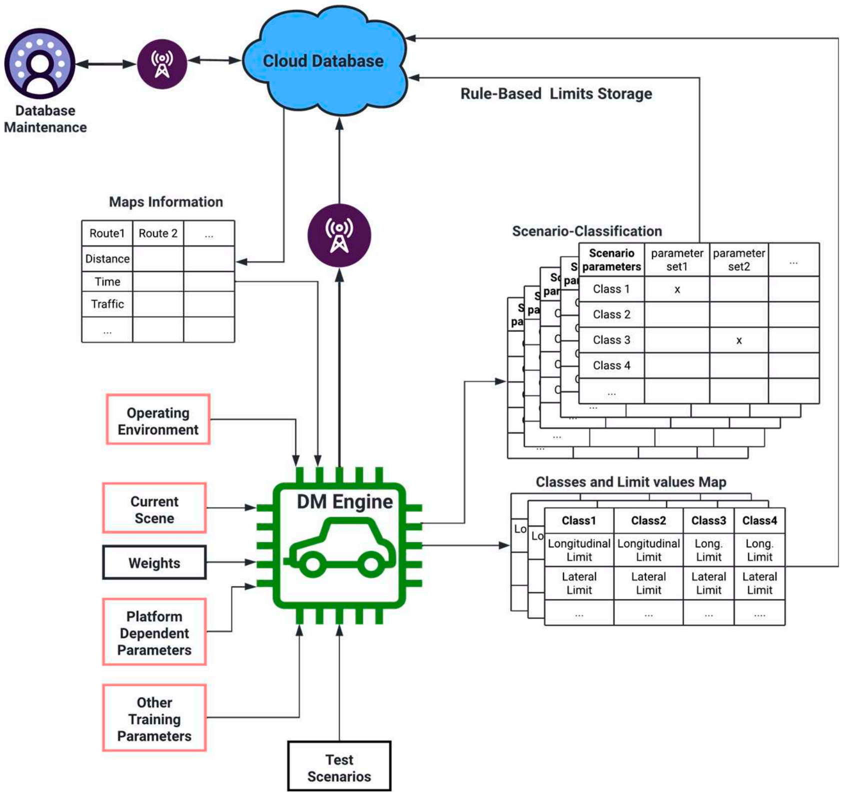

Figure 3. DM Engine and the Cloud Database Interactions[V2I].

To utilize a DM algorithm, identifying the parameters and tuning hyper parameters is crucial. These parameters encompass operating environment factors (weather, lighting, connectivity), current scene details (road geometry, conditions, markings, signs, structures, actors, ego vehicle parameters), and platform-dependent parameters (ego vehicle system capabilities and physical limitations) as shown in Figure 4. These considerations, coupled with criticality metrics, such as trajectory, maneuver, energy, uncertainty, or their combinations, contribute to effective DM algorithm performance, as highlighted by Cai et al. [17].

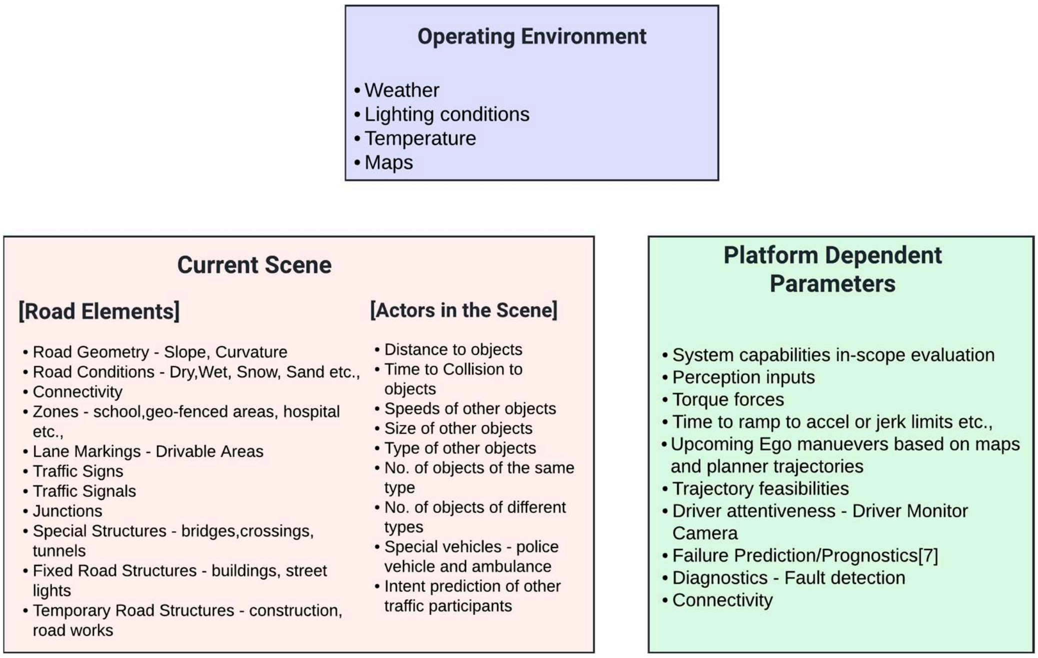
Operating Environment
• Weather
• Lighting conditions
• Temperature
• Maps
Current Scene
[Road Elements]
• Road Geometry - Slope, Curvature
• Road Conditions - Dry,Wet, Snow, Sand etc.,
• Connectivity
• Zones - school,geo-fenced areas, hospital etc.,
• Lane Markings - Drivable Areas
• Traffic Signs
• Traffic Signals
• Junctions
• Special Structures - bridges,crossings, tunnels
• Fixed Road Structures - buildings, street lights
• Temporary Road Structures - construction, road works
[Actors in the Scene]
• Distance to objects
• Time to Collision to objects
• Speeds of other objects
• Size of other objects
• Type of other objects
• No. of objects of the same type
• No. of objects of different types
• Special vehicles - police vehicle and ambulance
• Intent prediction of other traffic participants
Platform Dependent Parameters
• System capabilities in-scope evaluation
• Perception inputs
• Torque forces
• Time to ramp to accel or jerk limits etc.,
• Upcoming Ego manuevers based on maps and planner trajectories
• Trajectory feasibilities
• Driver attentiveness - Driver Monitor Camera
• Failure Prediction/Prognostics[7]
• Diagnostics - Fault detection
• Connectivity

Figure 4. Scenario Identification inputs for DM Algorithms.

3. Limits Classification

The autonomy system's primary functions at different layers can be explained in simple steps as follows and has been illustrated in Figure 1:

Step1: Perception detects, predicts and tracks the changes in the first 2 categories of scenario inputs mentioned above in Figure 4. [*At certain Accuracy and Reliability levels*].

Step2: Planner [10] based on current data, previous data, previously followed behav- iors and checking the intended behaviors against traffic rules and violations determines a set of feasible trajectories. **Step2.1:** Feasibility checker algorithm within the Planner considers the 3 categories of inputs like the perception inputs, environmental conditions and in-vehicle parameters to pick and choose the limits to be applied and picks the most feasible and appropriate trajectory - [*This is the algorithm that is being further expanded to add more DM capabilities and the cloud database support in this paper to create limits tables and choose the limits dynamically*.]

Step3: This is in turn updated in the limits database on the cloud using V2I [3]. This limits database acts as a ground truth in the known scenarios, backup in some extreme cases and failure cases where the DM Engine's capabilities are limited. Some corner cases [17] scenario limits or tables are created as well before-hand to provide more coverage.

Step4: Controller [21],[34] follows the most feasible and appropriate trajectory selected by the planner, by applying the limits that were chosen by the Feasibility checker algorithm.

Step5: Safety Controller checks if there are limits applied and if they are within the safe operational boundaries. If not, a pre-coded set of limits will be applied. [*These limits are present on-board the vehicle that are pre-loaded using the cloud database from Step4.*]

The feasibility checker algorithm in Step 3, illustrated in Figure 4, will incorporate DM algorithms to automatically select appropriate limits for each scenario. The ML algorithm classifies scenarios into four distinct classes based on input parameters. Focusing on longitudinal and lateral control limits, Figure 5 depicts a clear classification of limits according to scenario severity and risk level. Three types of limits—platform, safety, and comfort—are applicable in all scenarios and further divided across the four classes:

1. ***Class 1:*** For this class, being at the highest severity and risk level [36], the maximum limits that the vehicle platform can allow needs to be applied. There are cases where the component designs can be altered as well to apply limits exceeding the current physical limitations [23]. Most of the critical scenarios [25] fall into this category.
2. ***Class 2:*** This class is still at a high risk and severity level but less than Class1. Choosing the limits that are within the platform limits but more than a normal safe operational limits would be necessary [22],[24].
3. ***Class 3:*** This class covers the most common scenarios that are seen during normal driving but high speed or other slightly elevated risk level conditions are associated with it. Choosing the limits that are within the safe operational constraints would be sufficient in this case.
4. ***Class 4:*** This class consists of all the normal driving scenarios with minimum possible risk level involved and applying the comfort limits is usually sufficient

While these classes are defined distinctly, there may be overlap in limit values based on operating scenarios. For instance, the least comfortable longitudinal limit in Class 4 could be the same as the best-case scenario within Class 3, indicating potential correlations between classes as shown in Figure 13.

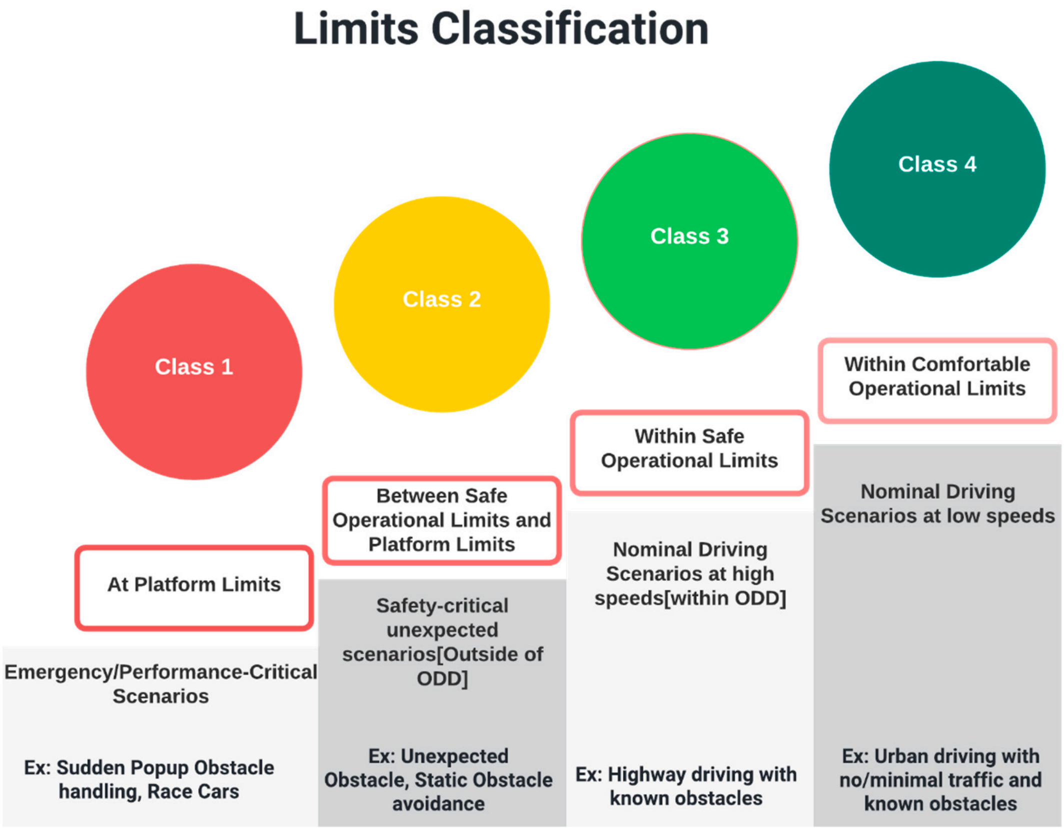
Limits Classification
Class 1
Class 2
Class 3
Class 4
At Platform Limits
Between Safe Operational Limits and Platform Limits
Within Safe Operational Limits
Within Comfortable Operational Limits
Emergency/Performance-Critical Scenarios
Safety-critical unexpected scenarios[Outside of ODD]
Nominal Driving Scenarios at high speeds[within ODD]
Nominal Driving Scenarios at low speeds
Ex: Sudden Popup Obstacle handling, Race Cars
Ex: Unexpected Obstacle, Static Obstacle avoidance
Ex: Highway driving with known obstacles
Ex: Urban driving with no/minimal traffic and known obstacles

Figure 5. Limits Classification.

4. Scenario Case Studies

Diverse case studies have been conducted using 2 different vehicle platforms to make sure that the classification of scenarios applies independent of the platform being tested. First vehicle platform is a full autonomy intent platform that was purpose built to scale to Level 5 Autonomy in the future. Second vehicle platform is an automated driving platform that was custom built to have Dual-Cockpit design for specific research use cases in terms of Advanced Driver Assistant Systems and Parallel

Autonomy [15],[46]. These 2 vehicle platforms were tested in various scenarios and environments like closed courses and public roads which provided insights into the distinct ranges of limits values applicable to each scenario class. As the scenario set could be very large to reach an accurate conclusion, we had to restrict the case studies to certain known critical scenarios like strong emergency braking, sharp turns, J-turn etc., which provided enough insights on the value ranges of the limits needed for each scenario class. One example selection of these ranges has been presented in this section based on the case studies, extensive working knowledge on similar case studies along with a projection of the results to other scenarios that already fall into one of the 4 classes presented in Section 3.

4.1. Longitudinal Limits:

- **Scenario 1:** Ego vehicle accelerating to 38 kmph and strong emergency braking toprevent crashing into the soft target car. In this case, the PCS feature has been disabled to test the underlying autonomy system's ability to decelerate at higher values close to the platform operating limits.

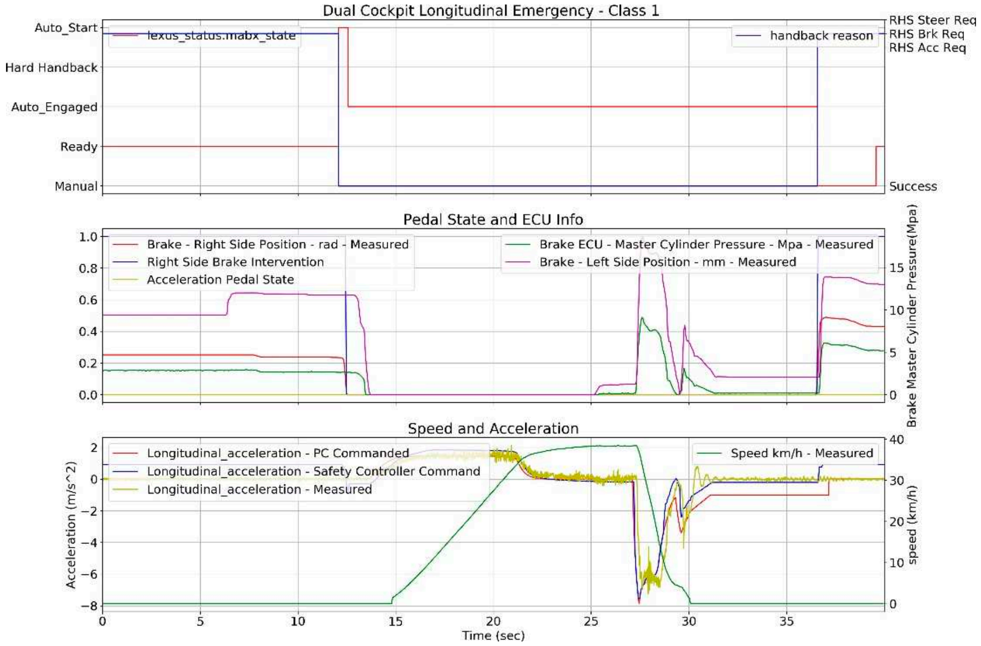

Figure 6. Strong Emergency Braking.

Results Discussion: The first subplot indicates the state of operation of the system like man- ual, hand back or autonomy. The second subplot shows the brake master cylinder pressure, if there was a brake intervention from the human, and the brake pedal position. Finally, the third subplot illustrates the deceleration profile of the system with the commanded, measured and the safety controller acceleration and deceleration values. At time instance 18-22 secs, the acceleration value of 1.7 m/s2 has been reached. At time instance 26-28 secs, the deceleration value of 0.6g has been reached. Also, the measured values follow the commanded values as expected. ***Conclusion:*** Deceleration Class1 example scenario at 5.88 m/s2 and an Acceleration Class 2 example scenario at 1.7 m/s2 have been identified.

- **Scenario 2:** Ego vehicle driving near a bicyclist at 35 kmph. This is a scenario where the safe operational limits need to be applied.

  Results Discussion: The maximum acceleration applied in this case did not exceed the range of 1-1.5 m/s2 acceleration. To be able to maintain the controllability of the vehicle in terms of getting closer to the bicyclist. ***Conclusion:*** Acceleration Class 3 at >1.5 m/s2 scenario has been identified.

- **Scenario 3:** Ego vehicle starting from stop scenario to reach 40 kmph where large acceleration value has been applied at the stop light and then proceeding through light traffic conditions at lower speeds.

Results Discussion: Between 15 - 17.5 secs, the acceleration value exceeded 2.0m/s2 which could be classified as the Acceleration Class1 value and then it follows nominal acceleration values of 0.7 to 0.0 m/s2 between 17.5 -20 secs which falls into the Class 4. Between 20-21 secs, nominal deceleration value of 0.2 m/s2 has been observed which falls into the Class 4 deceleration values category. ***Conclusion:*** Acceleration Class 1 and 4 along with Deceleration Class 4 scenarios have been identified.

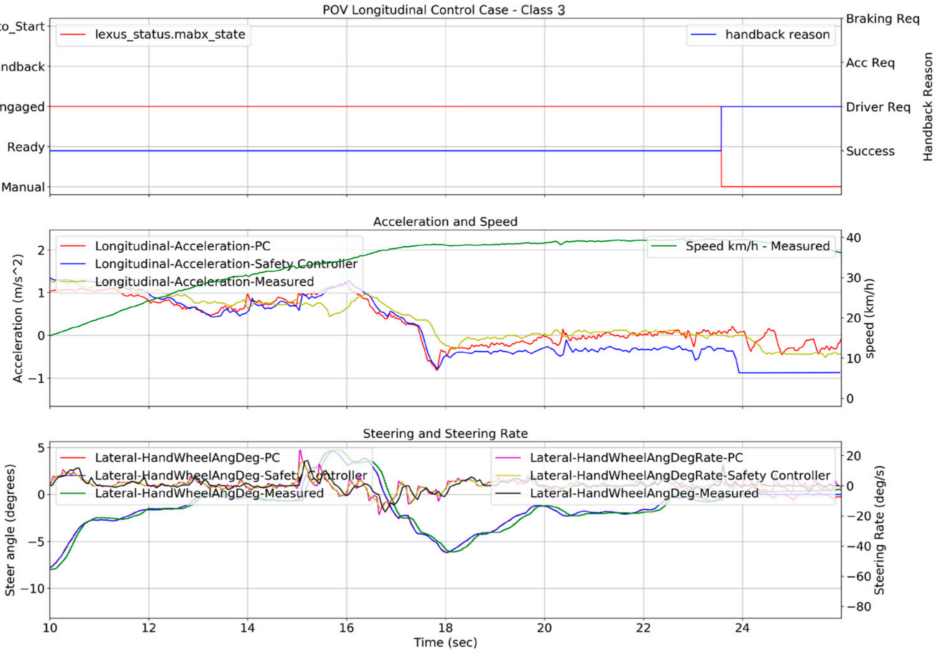

Figure 7. Driving near a Bicyclist.

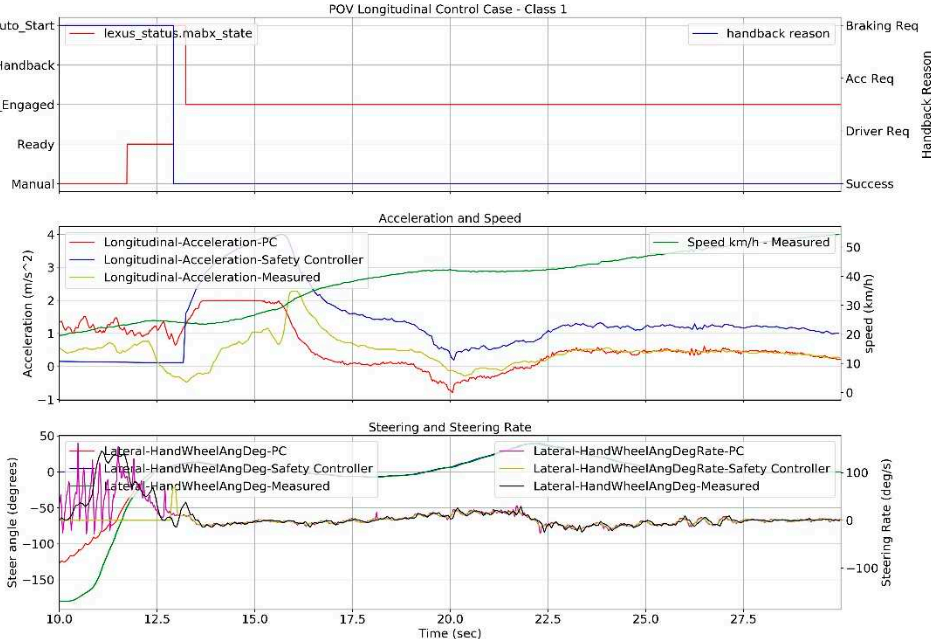

Figure 8. Start From Stop.

4.2. Lateral Limits:

- **Scenario 1:** Ego vehicle making a sharp right turn at an intersection with an angled right turn that has a sudden steep curvature to follow the mapped route.

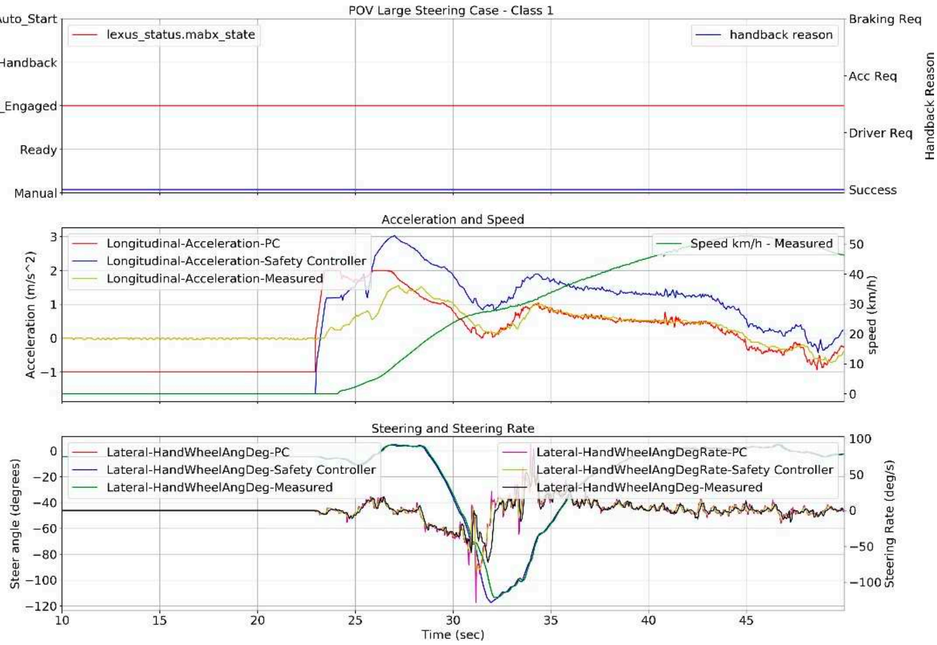

Figure 9. Sharp Turn.

Results Discussion: Between 30 - 35 secs, there is a large steering angle between 100-120 deg which falls into the Class 1 steering angle values. At 32 secs, there is a large steering angle rate of 80 deg/sec which falls into the Class 3 of the steering angle rate. ***Conclusion:*** Steering angle Class1 and steering angle rate Class 3 scenarios have been identified.

- **Scenario 2:** Ego Nudging in-lane towards left for bicyclist on the right side.

Results Discussion: This is a drive-by-wire system example and we have two steering angles i.e a Dual-Cockpit vehicle. The 2nd subplot plot: blue line is the measured steering angle from the hand wheel which reflects the driver's input. The purple line is the calculated steering angle from the road wheel, which means the AV stack's input. The 3rd plot: blue line is the measured steering angle rate from the hand wheel, which reflects the driver's input. The red line is the calculated steering angle from the road wheel, which means the AV stack's input.The blue line and red line are not necessarily matching, because of the the driver's input. We see that the blue line in the 2nd subplot represents a small steering angle around 8 deg to the left being applied for nudging between 14-20 secs. ***Conclusion:*** Steering angle Class4 scenario has been identified.

- **Scenario 3:** Ego making a J-turn at 20 kmph

Results Discussion: On the 2nd subplot, red line indicates the steering angle commanded from the PC.Between 37-43 secs, the steering angle commanded >300 deg to be able to complete the J-turn maneuver at 20kmph. ***Conclusion:*** Steering angle Class2 scenario has been identified.(Note:Based on lower speeds like 20kmph, the limit for Class2 would be <350 deg)

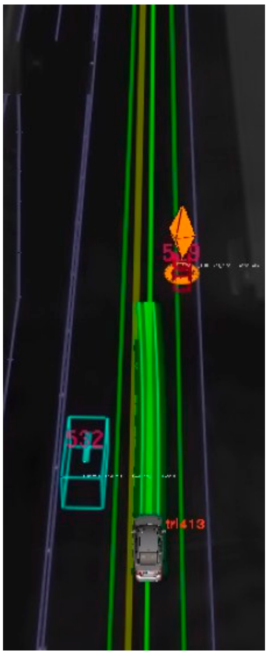

Figure 10. Nudging Scenario.

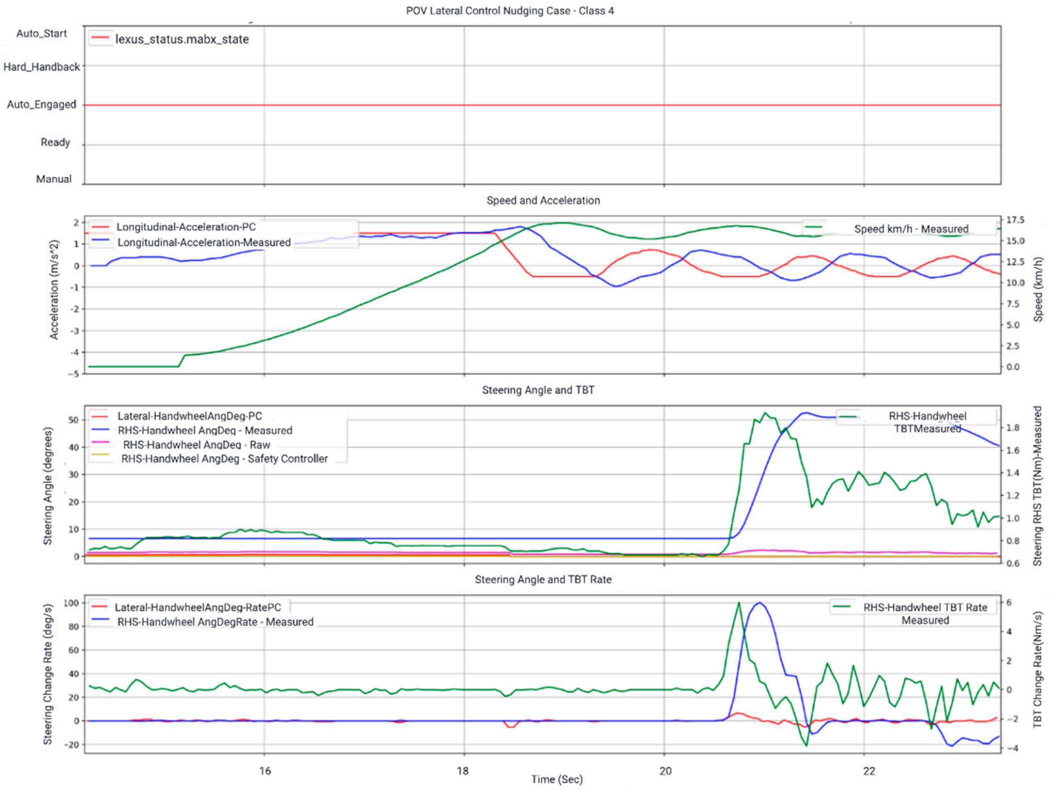

Figure 11. Nudging.

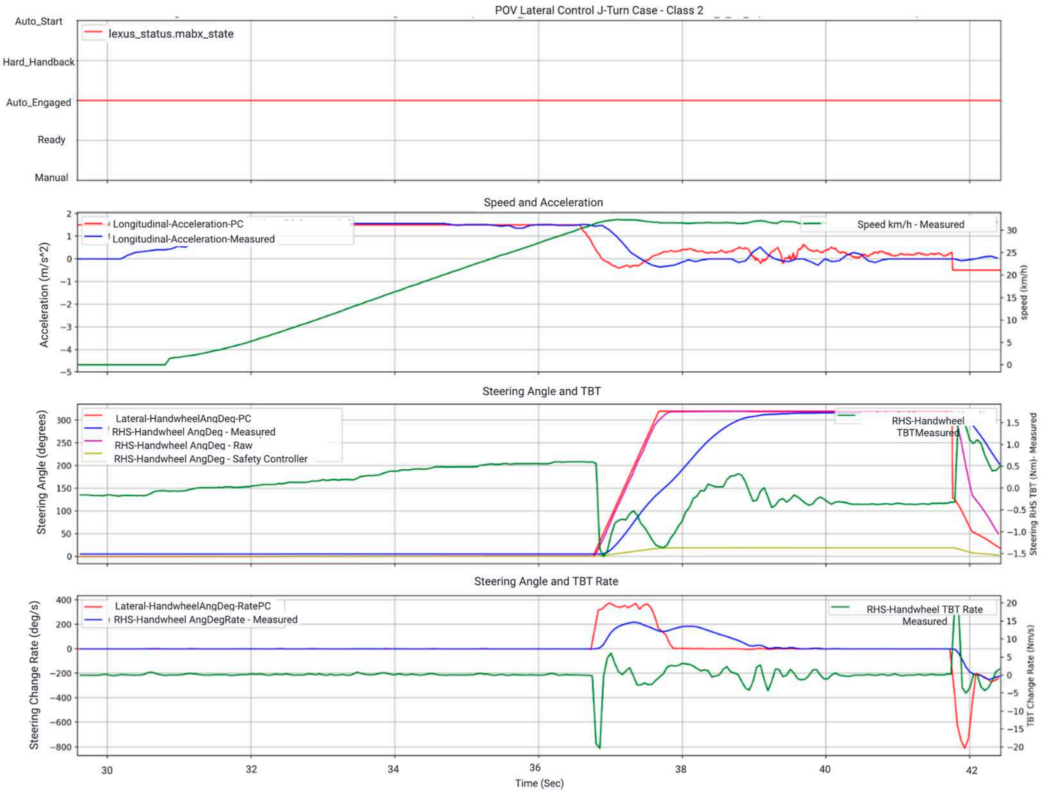

Figure 12. J-Turn.

4.3. More Class Examples:

Class 1:[Emergency Cases]

- As per Funke J. et. al [7] emergency scenarios demand vehicle maneuvers at operational limits to prevent collisions and maintain stability.
- Extreme cases, such as race car driving, exemplify system models operating at friction limits.
- Longitudinal critical scenarios include rapid acceleration from a stop or reaching a specific speed promptly.
- Lateral critical scenarios involve U-turn steering and swerving to avoid sudden obstacles.
- Operating at or above friction limits on icy/snowy roads is another example in this class.

Class 2:[Warning Cases]

- Longitudinal: Braking for Lead Vehicle, Worst-case scenario considers a suddenly stopped lead vehicle scenario.
- Lateral: Nudging over parked cars, U-turn scenarios etc.,

Class 3:[Elevated Caution]

- Longitudinal :Coming to a stop at the Traffic lights stop line, Normal braking for lead vehicle scenarios etc.,
- Lateral : Taking Right Turns at the Traffic Lights at lower speeds.

Class 4:[Nominal]

- Longitudinal: Low speed urban driving with less traffic, normal braking for speed adjustments during low speed driving.
- Lateral: Normal steering or no steering required scenarios.

Based on scenario case studies, prior working knowledge, and examples provided earlier, Tables 1 and 2 summarize chosen limit values for longitudinal and lateral limits, respectively. Table 3 provides an example of deceleration limit classification based on scenario inputs like the current scene, weather conditions and platform dependent parameters as was introduced in Section 2. Finally, Figure 13 illustrates the behavior of limit values when transitioning between classes, presenting variations of overlapping and non-overlapping classes.

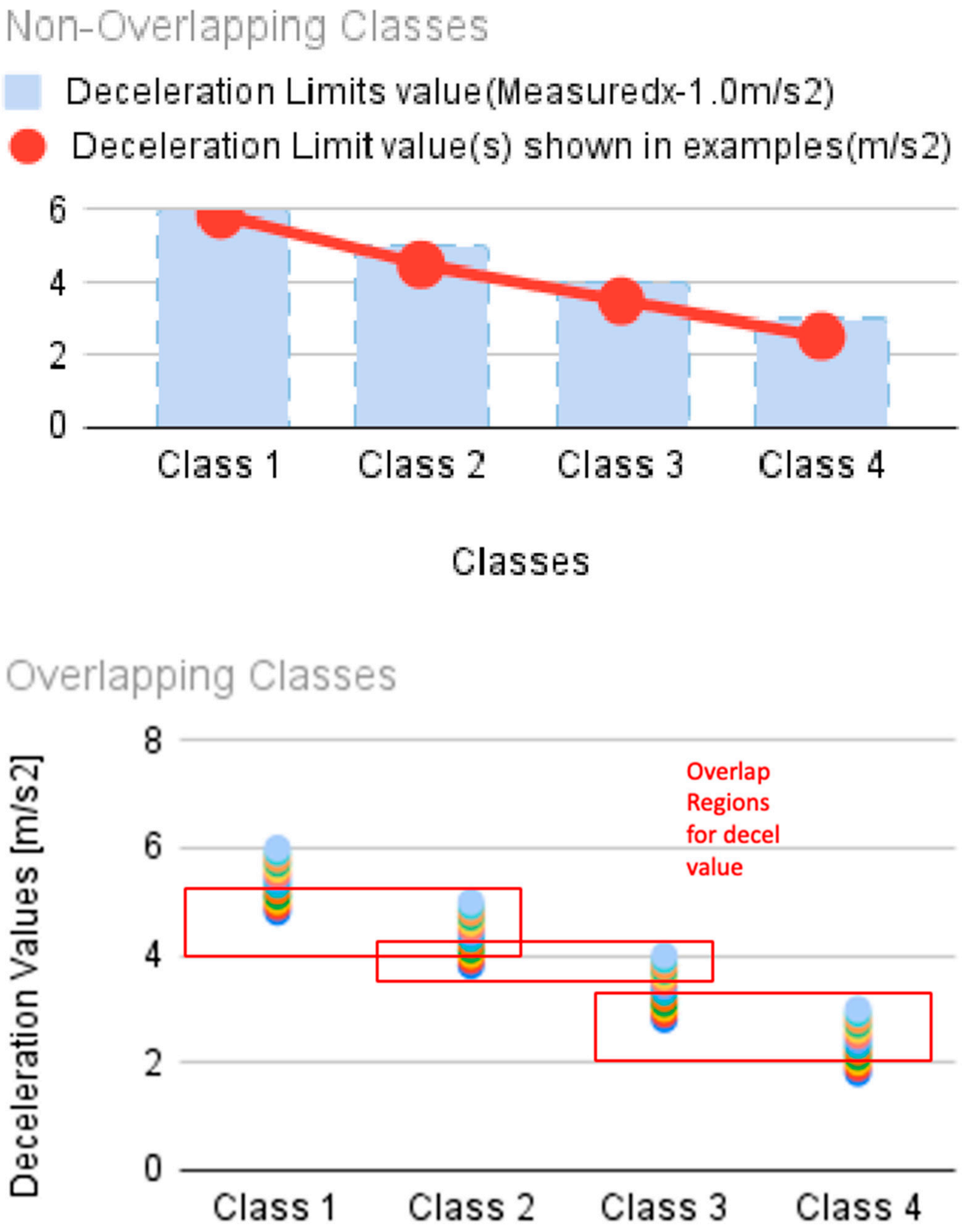

Figure 13. Overlapping and Non-Overlapping examples with deceleration limits values.

Table 1. Longitudinal Limit classification values Example.

Classes	**Deceleration Limits value (Measuredx-1.0m/s2)**	**Acceleration Limits value (Measured m/s2)**
Class 1	> 5.0	> 2.0
Class 2	4.0 - 5.0	1.5 - 2.0
Class 3	3.0 - 4.0	0.7 - 1.5
Class 4	0 - 3.0	0 - 0.7

Table 2. Lateral Limit classification values Example.

Classes	Steering Angle Limits value (Measured deg)	Steering Angle Rate Limits value (Measured deg/sec)	Vehicle Speed being considered[mph]
Class 1	> 100	> 120	45
Class 2	80-100	100-120	45
Class 3	50-80	75-100	45
Class 4	< 50	> 75	45

Table 3. Deceleration limits Classification Example based on Operating Environment, Current Scene and Platform dependent parameters.

ODD	Current Scene	Platform dependent	Scenario	Class
Snowy	very close cut-in vehicle	>5.0 m/s^2	Heavy Traffic	1
High Curvature	Oncoming high-speed traffic	>5.0 m/s^2	No lane markings	1
Dark	Pedestrian Approaching fast	>5.0 m/s^2	Brake for Pedestrian	1
Stop sign Intersection	Pop-up object in-lane	4.0-5.0 m/s^2	Ambulance Approaching	2
Signalized Intersection	Pop-up object in-lane	4.0-5.0 m/s^2	Green Traffic Light	2
Signalized Intersection	Other traffic has right of way too	4.0-5.0 m/s^2	Green Traffic Light	2
High Curvature	Motorcyclist Nearby High Speed	4.0-5.0 m/s^2	Brake for motorcyclist	2
Rainy	lead vehicle ahead	3.0-4.0 m/s^2	Brake for Lead vehicle	3
Day Light	Pedestrian Approaching	3.0-4.0 m/s^2	Brake for Pedestrian	3
Straight Road	Near parked cars	3.0-4.0 m/s^2	Low Speed	3
Sunny	no vehicles nearby	<3.0 m/s^2	Light Traffic	4
Signalized Intersection	Light Traffic	<3.0 m/s^2	Green Traffic Light	4

5. DM Algorithm Case Study

As we have now derived the limits tables and their mapping towards the classes***(Scenario Classes***<=>***Limits Tables)*** already in Section 4. This section provides a case study on how to apply a Decision Making algorithm using the Feasibility Checker within the Planner as defined in Section 3 in Step 2.1 to determine the "***Classes***" of the driving scenarios. For this case study, Multi-Criteria Decision Making(MCDM) algorithm [44] is being considered. MCDM is more suitable for complex decision problems with multiple conflicting criteria. It is highly flexible, offers clear interpretation of criteria weights and alternatives' performance. MCDM is really good at handling uncertainty through sensitivity analysis and scenario analysis. There are multiple types of MCDM algorithms [44] like WSM, AHP, WPM and TOPSIS [45]. TOPSIS algorithm has been chosen for our case study as it is relatively simple, easy to understand and can be easily adapted to different decision making scenarios by adjusting criteria and their weights. We considered one example scenario of "**Driving in Rain**" and created a synthetic randomly generated dataset as shown in Table 4 that consisted of 1000 different scenarios with multiple variations of parameters like Rain level, Time to Collision (TTC) to other objects in the current scene, Ego vehicle speed in the current scene and the road surface friction. The Rainfall rates have been taken from the AVSC ODD Framework [12] i.e., Light rain: Less than 0.1 in/h, Moderate rain: 0.1 to 0.3 in/h, Heavy rain: 0.3 to 2.0 in/h and Very heavy rain: More than 2.0 in/h. Ego vehicle speeds have been considered between 5 and 85 mph which covers a wide range of speeds. Friction values for Dry Asphalt or Concrete (0.7 to 0.9), Wet Asphalt or Concrete (0.4 to 0.7) and other low friction values have been considered upto a value of 0.0. For the ease of TOPSIS application, we have used equal weights for all parameters to start with.

The diagonal cells in Figure 14 contain histograms that represent the distribution of values for each variable. Histograms display the frequency or count of data points within specific bins or intervals. The shape of the histogram provides insights into the distribution of a single variable. This is important in exploratory data analysis (EDA) to identify patterns and potential issues with individual variables. In the scatterplot matrix's off-diagonal cells, each point represents a data point, with its position on the x and y axes corresponding to values of the two variables being compared. Random scattering of points suggests a weak or no correlation between the variables and hence with the scenario set being considered.

doi:10.20944/preprints202401.2034.v1

Table 4. [1000 rows x 4 columns] 1000 Scenario Dataset example for Driving in Rain using random generation.

No.	RainFall	TTC_Value	Vehicle_Speed	Friction
0	2.556383	46.289444	61.887093	0.756362
1	2.610514	27.868715	61.879226	0.779864
2	4.664956	57.172142	7.371099	0.487338
3	3.147938	48.526789	84.401135	0.664017
4	1.485411	53.829917	18.872669	0.021287
...	...	...	...	...
995	1.814006	49.973002	9.751570	0.325342
996	4.216090	10.041586	21.248852	0.573953
997	2.281334	2.411490	61.348333	0.696512
998	3.541673	35.457781	45.328620	0.348215
999	0.493485	15.738346	63.436905	0.869108

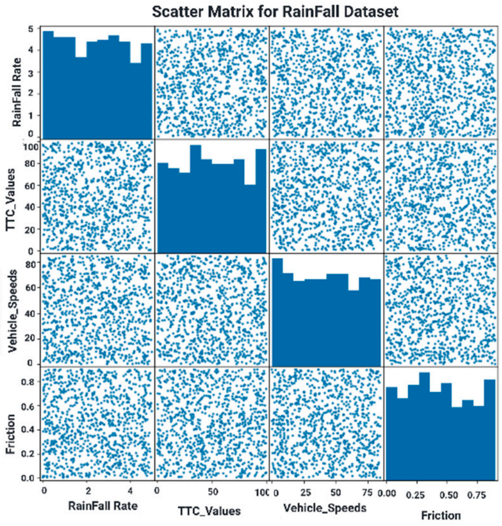

Figure 14. Scatter Matrix for the created RainFall Dataset.

5.1. TOPSIS Algorithm:

The generated dataset has been considered as the matrix X with n scenario variations with m criteria. First step is to normalize each criteria and then assign the weights to each criteria.

$$\| X_{ij} = \frac{X_{ij}}{\sqrt{\sum_i^n X_{ij}^2}}$$

$$V_{ij} = \| X_{ij} * W_j$$

Then, determine the ideal $^{(V_j^+)}$ and non-ideal $^{(V_j^-)}$ values for each criteria from the previously normalized and weighted values. These values are then used to calculate the Euclidean distances to each criteria values in the dataset.

$$S_i^+ = \sqrt{\sum_j^m (V_{ij} - V_j^+)^2}$$

$$S_i^- = \sqrt{\sum_j^m (V_{ij} - V_j^-)^2}$$

Final step is to calculate the relative performance score for each scenario within the dataset.

doi:10.20944/preprints202401.2034.v1

$$P_i^- = \frac{S_i^-}{S_i^- + S_i^+}$$

Based on the relative performance scores, the range of these scores have been equally distributed into 4 different classes with specified ranges resulting in the classified dataset with the corresponding classes assigned as shown in Table 5. Using this algorithm, out of the 1000 scenarios, there were 76 scenarios identified as Class1, 368 scenarios identified as Class2, 453 scenarios identified as Class3 and 103 scenarios identified as Class4.

Table 5. [1000 rows x 5 columns] Example for Classified Driving in Rain 1000 Scenario Dataset.

No.	RainFall	TTC_Value	Vehicle_Speed	Friction	Class
0	2.556383	46.289444	61.887093	0.756362	3.0
1	2.610514	27.868715	61.879226	0.779864	3.0
2	4.664956	57.172142	7.371099	0.487338	1.0
3	3.147938	48.526789	84.401135	0.664017	3.0
4	1.485411	53.829917	18.872669	0.021287	2.0
...	...	...	...	...	...
995	1.814006	49.973002	9.751570	0.325342	2.0
996	4.216090	10.041586	21.248852	0.573953	2.0
997	2.281334	2.411490	61.348333	0.696512	4.0
998	3.541673	35.457781	45.328620	0.348215	3.0
999	0.493485	15.738346	63.436905	0.869108	4.0

Figure 15 represents a 5-dimensional scatter plot with all 4 criteria plotted against the classes that have been determined using the TOPSIS algorithm. X-axis represents the RainFall Rates in inches/hour, Y-axis represents the TTC values, Z-axis represents the Ego vehicle speed values, size of the scatter plot data points represent the Friction values, color of the scatter plot data points represents the classes and the corresponding heat map has been provided. This type of plot helps visualize the mapping of the multiple criteria and their values being considered in the generated data to their corresponding classes.

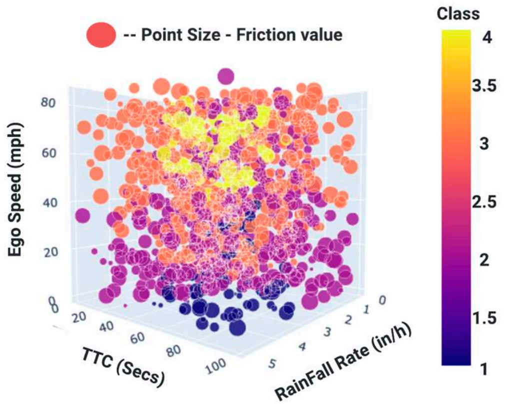

Figure 15. Classified Driving in Rain 1000 Scenario Dataset.

Hence, this example provides one way of classifying the scenarios using known MDCM method like TOPSIS. This work can be extended to use more Machine Learning (ML) and Deep Learning techniques to enhance the performance of the classification using decision trees and random forest algorithms.

6. Conclusion

This paper outlines a systematic approach to defining the Operational Design Domain (ODD), system capabilities, and scenarios, providing guidance for developing Level 2,3, and 4 autonomy systems. It traces the evolution of safety limits, identifies scenario classification parameters for the Decision-Making (DM) Engine, and describes the primary functions of each autonomy stack layer concerning safety limits application. The classification of safety limits into four categories is presented, offering a similar reference to ISO26262 severity or risk levels. The analysis includes real vehicle platforms and tests, informing the classification of actuation values into limits classes using scenario case studies from existing driving logs. These limits are stored in a cloud database, reducing on-board computing resources and promoting fleet-wide utilization. The paper concludes with an example DM algorithm for detecting "Driving in Rain" scenarios, demonstrating the applicability of the proposed classification framework to various scenarios and use cases in the industry's journey toward Level 5 Autonomy.

7. Future Work

MCDM algorithms could benefit from integration with Machine Learning (ML) and Deep Neural Networks (DNNs) to address increasing complexity with a growing number of criteria. For handling overlapping classes, which are beyond this paper's scope, fuzzy logic or DNNs might offer more effective solutions. Ongoing work explores the use of ML and DNNs, with results to be included in future publications.

To enhance the accessibility and accuracy of the cloud database, a connected vehicle concept using Vehicle-to-Infrastructure (V2I) communication can involve multiple Original Equipment Manufacturers (OEMs) and entities contributing and subscribing. Further developments should incorporate more ML and Deep Learning algorithms for precise scenario prediction and classification, aiming to refine the correlation between classes over time. Another potential extension involves creating driver models with varying degrees of passiveness or aggressiveness, aligning with human behavior studies, and mapping classes to different driving modes (normal, eco, sport) [40],[41]. These additions can introduce more granularity and versatility to the system adding the human machine interaction parameter set in determining the limits.

References

1. Johnson, Matthew C. "Development of a robotic vehicle control system." PhD diss., 2013.[Google Scholar][CrossRef]
2. C. Ma, J. Xue, Y. Liu, J. Yang, Y. Li and N. Zheng, "Data-Driven State-Increment Statistical Model and Its Application in Autonomous Driving," in IEEE Transactions on Intelligent Transportation Systems, vol. 19, no. 12, pp. 3872-3882, Dec. 2018, doi: 10.1109/TITS.2018.2797308. [Google Scholar][CrossRef]
3. I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro and E. Cerqueira, "Towards software-defined VANET: Architecture and services," 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), Piran, Slovenia, 2014, pp. 103-110, doi: 10.1109/MedHocNet.2014.6849111. [Google Scholar][CrossRef]
4. D. Garikapati and Y. Liu, "Dynamic Control Limits Application Strategy For Safety-Critical Autonomy Features," 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC), Macau, China, 2022, pp. 695-702, doi: 10.1109/ITSC55140.2022.9922214. [Google Scholar][CrossRef]
5. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles J3016_202104. J3016_202104: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles – SAE International. [CrossRef]
6. Hajiloo, R., Abroshan, M., Khajepour, A., Kasaiezadeh, A., & Chen, S. K. (2020). Integrated steering and differential braking for emergency collision avoidance in autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 22(5), 3167-3178. [Google Scholar][CrossRef]
7. J. Funke, M. Brown, S. M. Erlien and J. C. Gerdes, "Collision Avoidance and Stabilization for Autonomous Vehicles in Emergency Scenarios," in IEEE Transactions on Control Systems Technology, vol. 25, no. 4, pp. 1204-1216, July 2017, doi: 10.1109/TCST.2016.2599783. [Google Scholar][CrossRef]

8. L. Tang, G. J. Kacprzynski, K. Goebel, A. Saxena, B. Saha and G. Vachtsevanos, "Prognostics-enhanced Automated Contingency Management for advanced automated systems," 2008 International Conference on Prognostics and Health Management, Denver, CO, USA, 2008, pp. 1-9, doi: 10.1109/PHM.2008.4711448. [Google Scholar][CrossRef]
9. PAS 1883:2020 Operational Design Domain (ODD) taxonomy for an automated driving system (ADS) – Specifi cation. Published by BSI Standards Limited 2020. ISBN 978 0 539 06735 4 ICS 03.220.20; 35.240.60. [CrossRef]
10. Schwarting, Wilko and Alonso-Mora, Javier and Rus, Daniela, "Planning and Decision-Making for Autonomous Vehicles,Annual Review of Control", Robotics, and Autonomous Systems,1(1),187-210,2018,doi: 10.1146/annurev-control-060117-105157. [Google Scholar][CrossRef]
11. ISO/CD PAS 8800 - Road Vehicles — Safety and artificial intelligence. [CrossRef]
12. AVSC00002202004: AVSC Best Practice for Describing an Operational Design Domain: Conceptual Framework and Lexicon – SAE International. [CrossRef]
13. ISO 34503:2023, Road Vehicles Test scenarios for automated driving systems Specification for operational design domain. [CrossRef]
14. U.S. Department of Transportation, National Highway Traffic Safety Administration, 2018, "A Framework for Automated Driving System Testable Cases and Scenarios", DOT HS 812 623. [CrossRef]
15. J. Dawson and D. Garikapati, "Extending ISO26262 to an Operationally Complex System," 2021 IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, 2021, pp. 1-7, doi: 10.1109/SysCon48628.2021.9447146. [Google Scholar][CrossRef]
16. L. Sedghi, Z. Ijaz, M. Noor-A-Rahim, K. Witheephanich and D. Pesch, "Machine Learning in Event-Triggered Control: Recent Advances and Open Issues," in IEEE Access, vol. 10, pp. 74671-74690, 2022, doi: 10.1109/ACCESS.2022.3191343. [Google Scholar][CrossRef]
17. J. Zhou and J. Beyerer, "Corner Cases in Data-Driven Automated Driving: Definitions, Properties and Solutions," 2023 IEEE Intelligent Vehicles Symposium (IV), Anchorage, AK, USA, 2023, pp. 1-8, doi: 10.1109/IV55152.2023.10186558. [Google Scholar][CrossRef]
18. Cai, J.; Deng, W.; Guang, H.; Wang, Y.; Li, J.; Ding, J. A Survey on Data-Driven Scenario Generation for Automated Vehicle Testing. Machines 2022, 10, 1101. doi: 10.3390/machines10111101. [Google Scholar][CrossRef]
19. Hendrik.A.Reese, Driving ! = Driving: Why operational design domain and scenario definition is the essence for safe autonomous driving systems. [CrossRef]
20. Philip Koopman, William.H.Widen, "Safety Ethics for Design and Test of Automated Driving Features",IEEE Design Test, Forthcoming, University of Miami Legal Studies Research Paper No. 4465034. [Google Scholar][CrossRef]
21. Elliot Weiss, J. Christian Gerdes, "Follow My Lead: Designing an ADAS that Shares Decision Making and Control with the Driver", 2023 IEEE Conference on Systems, Man, and Cybernetics (SMC), October, 2023. [CrossRef]
22. John C. Kegelman, Lene K. Harbott & J. Christian Gerdes (2017) Insights into vehicle trajectories at the handling limits: analysing open data from race car drivers, Vehicle System Dynamics, 55:2, 191-207, DOI: 10.1080/00423114.2016.1249893. [Google Scholar][CrossRef]
23. Goh, J., Gerdes, J. C., Bobier-Tiu, C., Pavone, M., Rock, S. M., & Stanford University. (2019). Automated vehicle control beyond the stability limits. [CrossRef]
24. FUNKE, J., GERDES, J. C., MITIGUY, P., & PAVONE, M. (2015). Collision avoidance up to the handling limits for autonomous vehicles. [CrossRef]
25. Song, Q., Tan, K., Runeson, P. et al. Critical scenario identification for realistic testing of autonomous driving systems. Software Qual J 31, 441–469 (2023). doi.org: 10.1007/s11219-022-09604-2. [Google Scholar][CrossRef]
26. Yuen KF, Chua G, Wang X, Ma F, Li KX. Understanding Public Acceptance of Autonomous Vehicles Using the Theory of Planned Behaviour. Int J Environ Res Public Health. 2020 Jun 19;17(12):4419. doi: 10.3390/ijerph17124419. PMID: 32575549; PMCID: PMC7345617. [CrossRef]
27. Sitinjak C, Tahir Z, Toriman ME, Lyndon N, Simic V, Musselwhite C, Simanullang WF, Hamzah FM. Assessing Public Acceptance of Autonomous Vehicles for Smart and Sustainable Public Transportation in Urban Areas: A Case Study of Jakarta, Indonesia. Sustainability. 2023; 15(9):7445. doi: 10.3390/su15097445. [CrossRef]
28. H. Martin, B. Winkler, S. Grubmüller and D. Watzenig, "Identification of performance limitations of sensing technologies for automated driving," 2019 IEEE International Conference on Connected Vehicles

and Expo (ICCVE), Graz, Austria, 2019, pp. 1-6, doi: 10.1109/ICCVE45908.2019.8965181. [CrossRef]

29. Yeong DJ, Velasco-Hernandez G, Barry J, Walsh J. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. Sensors. 2021; 21(6):2140. doi: 10.3390/s21062140. [CrossRef]
30. Romuald Aufrère, Jay Gowdy, Christoph Mertz, Chuck Thorpe, Chieh-Chih Wang, Teruko Yata,Perception for collision avoid- ance and autonomous driving, Mechatronics,Volume 13, Issue 10,2003,Pages 1149-1161,ISSN 0957-4158, doi: 10.1016/S0957- 4158(03)00047-3. [CrossRef]
31. S. Riedmaier, T. Ponn, D. Ludwig, B. Schick and F. Diermeyer, "Survey on Scenario-Based Safety Assessment of Automated Vehicles," in IEEE Access, vol. 8, pp. 87456-87477, 2020, doi: 10.1109/ACCESS.2020.2993730. [CrossRef]
32. A. Rodionova, I. Alvarez, M. S. Elli, F. Oboril, J. Quast and R. Mangharam, "How safe is safe enough? Automatic Safety Constraints Boundary Estimation for Decision-Making in Automated Vehicles," 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, 2020, pp. 1457-1464, doi: 10.1109/IV47402.2020.9304756. [Google Scholar] [CrossRef]
33. Li, S., Yang, J., Chen, W., Chen, X. (2014). Disturbance Observer-Based Control: Methods and Applications. United Kingdom: Taylor & Francis. [CrossRef]
34. S. Kouro, P. Cortes, R. Vargas, U. Ammann and J. Rodriguez, "Model Predictive Control—A Simple and Powerful Method to Control Power Converters," in IEEE Transactions on Industrial Electronics, vol. 56, no. 6, pp. 1826-1838, June 2009, doi: 10.1109/TIE.2008.2008349. [CrossRef]
35. Koibuchi, K., Yamamoto, M., Fukada, Y., and Inagaki, S., "Vehicle Stability Control in Limit Cornering by Active Brake," SAE Technical Paper 960487, 1996, doi: 10.4271/960487. [CrossRef]
36. ISO 26262-1:2018 Road vehicles Functional safety, International Standard under systematic review [90.20], vol.2(33),ISO/TC 22/SC 32, ICS : 01.040.43 43.040.10. [CrossRef]
37. V. A. Laurense, J. Y. Goh and J. C. Gerdes, "Path-tracking for autonomous vehicles at the limit of friction," 2017 American Control Conference (ACC), Seattle, WA, USA, 2017, pp. 5586-5591, doi: 10.23919/ACC.2017.7963824. [CrossRef]
38. Alan S. Hoback, Relationships between aggressive driving behaviors, demographics and pareidolia, Transportation Research Interdisciplinary Perspectives, Volume 2, 2019, 100037, ISSN 2590-1982, doi: 10.1016/j.trip.2019.100037.[CrossRef]
39. Z. Wang et al., "Driver Behavior Modeling Using Game Engine and Real Vehicle: A Learning-Based Approach," in IEEE Transactions on Intelligent Vehicles, vol. 5, no. 4, pp. 738-749, Dec. 2020, doi: 10.1109/TIV.2020.2991948. [CrossRef]
40. Timo Melman, Joost de Winter, Xavier Mouton, Adriana Tapus & David Abbink (2021) ,"How do driving modes affect the vehicle's dynamic behaviour? Comparing Renault's Multi-Sense sport and comfort modes during on-road naturalistic driving", Vehicle System Dynamics, 59:4, 485-503, DOI: 10.1080/00423114.2019.1693049. [CrossRef]
41. Kenny Longdon, "Driving Modes Explained: What are They?", Evans Halshaw, Nov 2022. [CrossRef]
42. Qin, S. Joe, and Thomas A. Badgwell. "An overview of industrial model predictive control technology." *AIche symposium series*. Vol. 93. No. 316. New York, NY: American Institute of Chemical Engineers, 1971-c2002., 1997. [CrossRef]
43. Z. Liang, J. Zhao, B. Liu, Y. Wang and Z. Ding, "Velocity-Based Path Following Control for Autonomous Vehicles to Avoid Exceeding Road Friction Limits Using Sliding Mode Method," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 3, pp. 1947-1958, March 2022, doi: 10.1109/TITS.2020.3030087. [Google Scholar][CrossRef]
44. Triantaphyllou, E. (2000). Multi-Criteria Decision Making Methods. In: Multi-criteria Decision Making Methods: A Comparative Study. Applied Optimization, vol 44. Springer, Boston, MA. doi: 10.1007/978-1-4757-3157-6_2. [CrossRef]
45. D.L. Olson, Comparison of weights in TOPSIS models, Mathematical and Computer Modelling, Volume 40, Issues 7–8,2004,Pages 721-727,ISSN 0895-7177, doi: 10.1016/j.mcm.2004.10.003. [CrossRef]
46. Naser, Felix et al. "A Parallel Autonomy Research Platform." 2017 IEEE Intelligent Vehicles Symposium. 11-14 June, 2017, Redondo Beach, CA, USA, IEEE, 2017. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.