Thesis Topic: Design of Validation Parameter Space to validate automated driving algorithms

Objective – Design of evaluation system to validate the autonomous car modules for systematic development and upgrade of system modules.

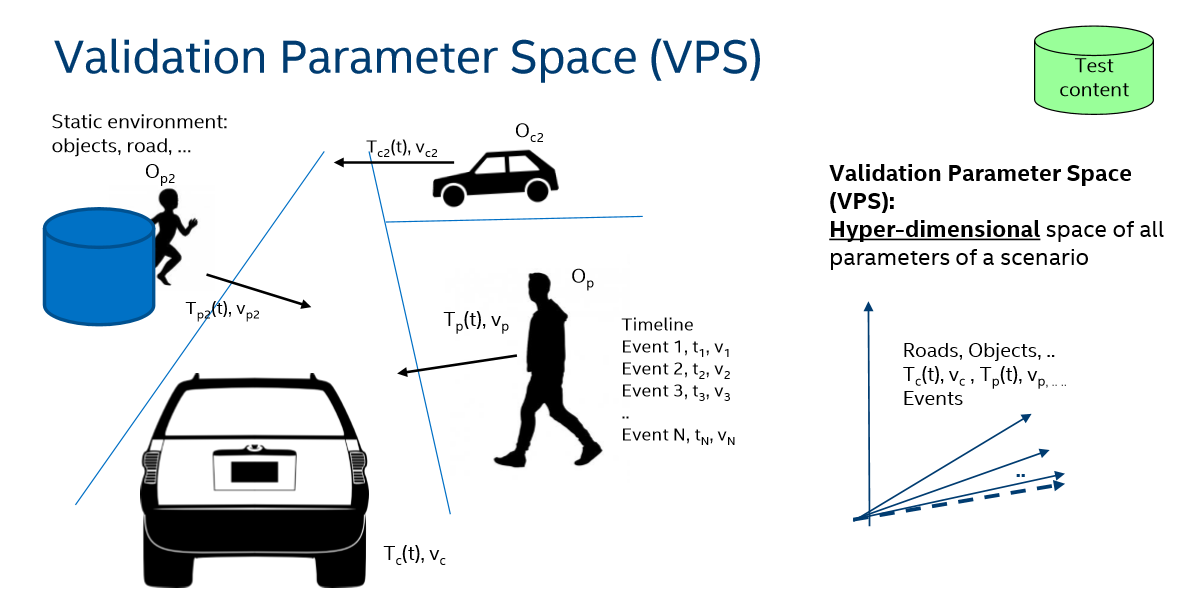


Figure 1. Validation parameter space (VPS)

Evaluation of any system is most important for continuous betterment and upgradation of the system. More complex the system is, more complex the evaluation should be. Modern cars are equipped with more than 150 ECU’s. The modules should be tested real time in synchronous with each ECU interacting with the every other module like perception and braking modules interaction. According to European regulations, an autonomous car5s should be evaluated at least for 240 million km. The interesting factor would be to question under what conditions one would like to the testing for 240 million km. Also the captured data cannot be changed to evaluate it under different conditions

For an autonomous car to be tested, Approved and intelligent methodologies to validate AV technologies. Environment surrounding it should be designed such that all the critical path of the algorithm should be rigorously evaluated and tested. The scene around the car should be designed with a particular parameters set to test the modules under most risky conditions. This parameter set is called as “**Validation Parameter Space” (S: {VPSi} i)**, i = dynamic and static objects like scene pedestrians, scene (buildings, weather, vegetation), other vehicles etc., S is state vector space of an object.

A risk function **r(S)** is derived out of the validation parameter space which defines the level of risk and criticality of the system. The function **r(S)** is 0, defining no risk and 1 when the system is exposed with highest possible risk. The system is exposed to the test conditions derived from **r(S)** to test the vulnerability of the algorithms and modules in check. Deriving the right **VPS(S)** with the desired **r(S)**, is a classic optimization/search problem. This function is a vast world of hyper dimensionality. The problem could be probed by starting with one part of the environment. For example, all the related objects in a scene could be mocked up except the object of interest i.e. illumination of the scene or the speed of the dynamic objects surrounding the car. The probabilistic relation between all the contents in the scene with the risk factor is studied to generate a dense hyper dimensional model. The desired scene is then extracted by sampling the required specifications.

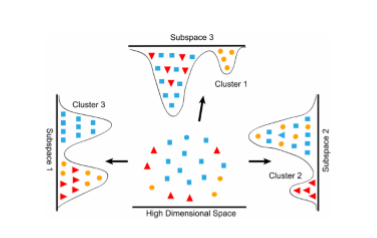


Figure 2. Hyper Dimension Space

A target function is a complementary function of risk function which validates the functionality of the module. Therefore target function should be first defined before structuring the risk function. With these functions. An apt scene will be created by combining the probabilistic models of these functions.

Approach:

1. Content creation – Blender (or unreal, unity) could be used to create the content and export it to the simulation engine like Carla. Content is parametrized.
2. Feed in relationship model of all the objects in scene with respect to each other (behavior)
3. Implement the search algorithm to find many combination of risk function r(S).
4. Render the created content with required quality.
5. Simulator – Test the content created in simulator.

Relationship model: This defines the interaction between the 2 objects at different dimensions. Since the goal is safety, we could look for KPI’s (factors) which influences risk highly.

**Factors to deal with :**

Functional Safety;

Risk of NN’s is low;

Perturbation influenced scenes