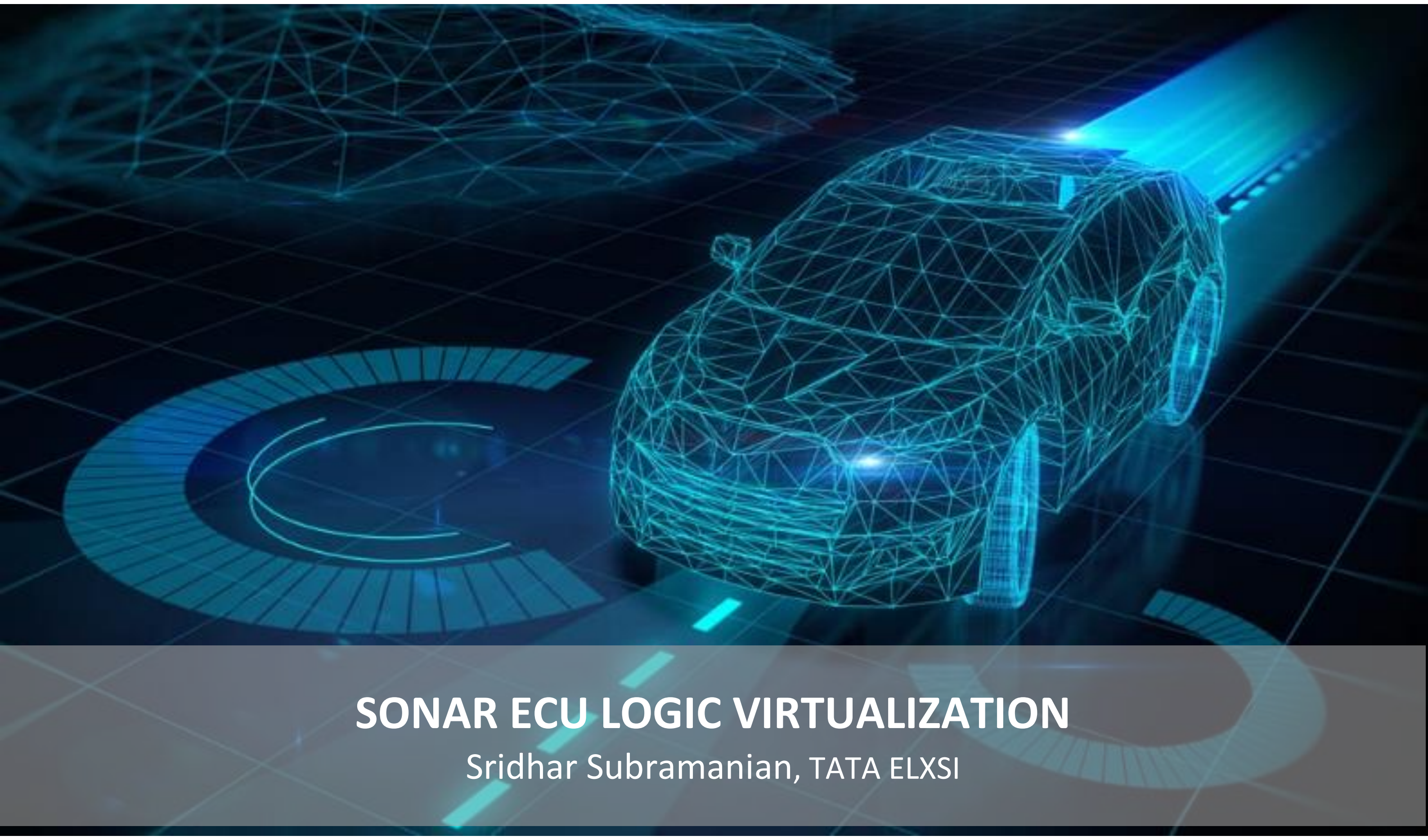


SONAR ECU LOGIC VIRTUALIZATION

Sridhar Subramanian, TATA ELXSI

TABLE OF CONTENTS

ABSTRACT

Vehicle Tests and real-world testing for AD/ADAS ECUs are absolutely necessary. Then again, it's truly difficult to create millions of scenarios in the real-world, to detect erroneous software functioning which could cause fatal accidents in real-world scenarios, which is expensive too. Due to this necessity, thoroughly checking the behavior of autonomous ECUs under various circumstances and the difficulty in generating a plethora of real-world scenarios the virtual validation methods are getting prominent.

Automotive companies are already in the process of adopting lab validation methodologies for validating AD/ADAS ECUs together with synthetic scenario generation tools. However, real functionality testing in the lab using real sensors (real Sonar, Radar, etc.) requires elaborate setups and is limited to a few obstacle simulations only. Real Sonar, Radar ECUs may be integrated to the HIL setups to execute some basic tests related to network, diagnostics, etc. but detailed testing of AD/ADAS ECU fusion algorithms becomes difficult in the lab. As a solution to this issue, the industry is moving towards virtualization of the logic of such ECUs so that these "soft ECUs" can be deployed in a lab setup. It opens up the possibility to test a wide variety of scenarios without the need for actual ECUs or record/replay methods.

This whitepaper provides a view of the common challenges faced while developing a virtual Sonar ECU Logic and presents possible approaches to overcome those challenges.

INTRODUCTION

Every year, 1.2 million lives are lost to traffic crashes around the world, 94% of which is due to human error. Big software companies and Automobile OEMs are betting high on autonomous driving technologies as an effective solution to reduce accidents on road and improve safety and traffic conditions. AD also has the potential in providing transportation facilities to differently-abled people and saving valuable time and energy for the "Automobile Drivers" of today. AD and ADAS technology has revolutionized the automotive sector forcing the industry players to bring innovative solutions.

AD/ADAS MARKET OUTLOOK AND THE FUTURE OF ULTRASONIC SENSING TECHNOLOGIES

Fig. 1 below shows some key facts from the Autonomous vehicles and the related sensors market. It is expected that the Autonomous vehicles market will grow steadily in the next 15 years with a CAGR in the range of 20% to 40%, this will surely have a growth impact on the sensor markets too. It is predicted that immediately Camera + Radar + Ultrasonic sensor market will grow while the Industry will see prominent use of LiDAR systems in the latter part of the journey once the production costs decrease.

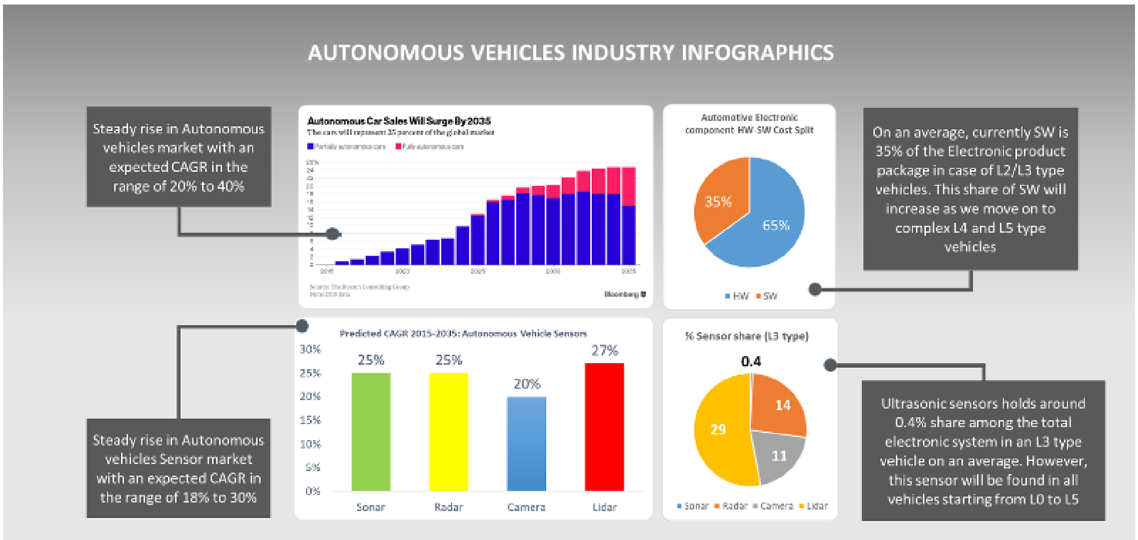


Figure 1: Some key statistics related to Autonomous Vehicles and Sensors market

Talking specifically about Ultrasonic sensing technologies, they hold just about 0.4% share among the total electronic systems in an L3 type vehicle. However, the use of Ultrasonic sensing technologies will continue throughout L0 to L5 vehicles and the data from this sensor will be fused with data from other sensors for sensing the surrounding environment, especially during the auto park process (Refer Fig 2). However small the percentage be, they will be an integral part of almost all kinds of Autonomous vehicles.

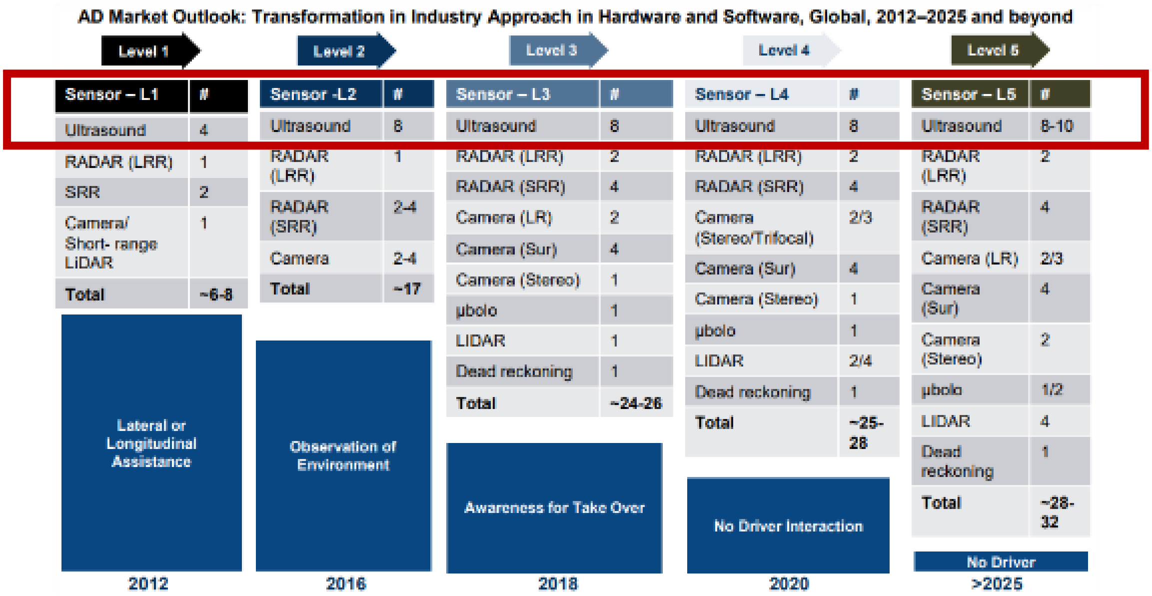


Figure 2: Ultrasonic sensor to remain an indispensable component of all types of Autonomous vehicles

TRANSFORMATION OF TESTING & VALIDATION METHODOLOGIES

A simple high-end car today has around 100 million SLOC (software lines of code) which will proliferate with the growth of autonomous features. And thus the amount of testing that is required for autonomous vehicles will also rise exponentially.

The sensors that are being used for AD/ADAS systems (e.g. Radar, Sonar, LiDAR, Camera, etc.) are far more advanced than the sensors like speed sensing GMR sensor or very simple resistive temperature sensors. They are so complex that a separate ECU is required to capture raw data and process it into meaningful chunks of data that can be passed on to other ECUs in the network. Finally, the AD / ADAS sensors sense the environment as a whole, unlike the sensors that were used to sense specific parameters like temperature, pressure or speed at a particular point. Hence testing Autonomous functions using real Radar/Sonar ECU in Lab is not practical (since creating different kinds of “real” scenarios in lab is much more difficult than creating a virtual scenario using simulation software like IPG Carmaker and dSPACE ASM).

These factors will urge the entire Automobile industry to raise the bar of the testing and validation processes. Vehicle testing or Lab testing in isolation will not suffice, the industry has to come up with novel ways to test each component that goes into such complex systems.

THE SHIFT TO VIRTUAL ECUs AND ECU LOGIC MODELS

The Automotive industry as a whole is now relying even more on virtual validation setups for AD/ADAS systems during the early product development phases to bring down cost and development time. Due to this, there is an increased requirement for virtualization of ECUs or ECU logics so that they can be integrated with the Lab test setups and simulation environment modeling tools to achieve the flexibility of testing a large number of virtual test scenarios in the lab before moving on to actual vehicle level testing. In the case of AD/ADAS Systems, few common ECUs/ECU logics that are being virtualized now are for the Ultrasonic ECU, RADAR ECU & CAMERA Modules.

CHALLENGES NORMALLY ENCOUNTERED IN CREATING SONAR MODEL

The following challenges may be encountered while creating a virtual Sonar ECU logic

CHALLENGE #1: HANDLING MULTIPLE COORDINATE SYSTEMS

When working with the virtual environment and related software tools, you may come across various coordinate systems as given in the table below

Coordinate system	General location of Origin	Use
Global Coordinate	Origin of the Scenario tool	Creating scenarios using roads, obstacles, ego car, etc.
Vehicle Coordinate	Centre of the front/rear/axles, etc.	Placement of vehicle components like tires, suspension, etc. to create a virtual vehicle model
Sensor Coordinate	Virtual sensor aperture	Raw data received for detected obstacles (distance/angle/etc.) is referenced to this coordinate system
OEM / Supplier Coordinate	Centre of the front/rear/axles etc.	This is the common coordinate system specified by the respective OEM/Supplier for standardization purposes

Table 1: Different coordinate systems and their uses

It is very essential to understand and use all these coordinate systems to minimize efforts that go into the design of the sensor model. Also, you must consider using different kind of coordinate transformation methods like translation, rotation, scaling, reflection, etc. to create a generic model that is independent of scenarios, vehicle variants, etc.

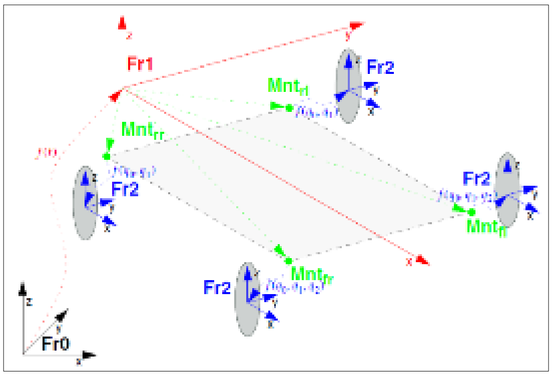


Figure 3: Global & Vehicle coordinate system

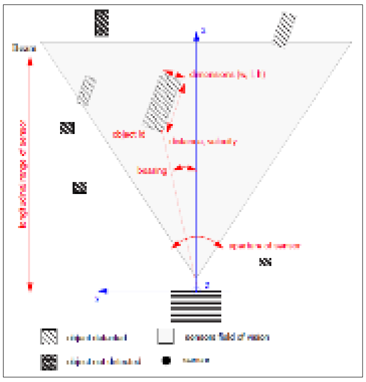


Figure 4: Sensor coordinate system

CHALLENGE #2: SONAR SENSOR PROFILING IN THE VIRTUAL ENVIRONMENT

The real-world sonar sensors do not have perfect circular/linear profiles/angles (horizontal and vertical). Also, the sensors mounted on the vehicle may or may not have an equal range and angle of transmission. In some cases, if radar is not used, sonar sensors around the car are calibrated with different ranges and angles to overlap with some regions that are usually covered by radar in an AD/ADAS system. Please refer Fig 5 below:

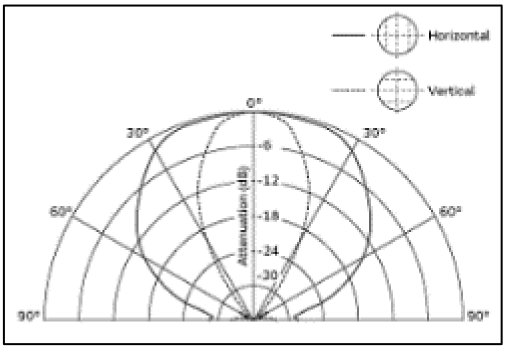


Figure 5: Murata MA58MF14-7N Ultrasonic Sensor profile data

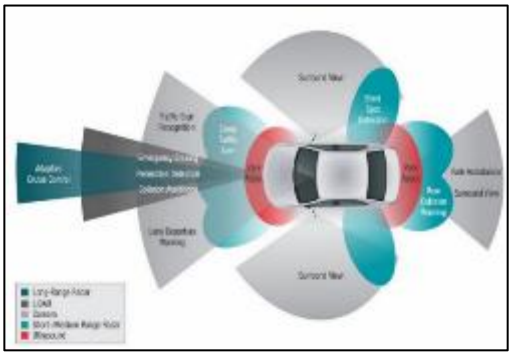


Figure 6: AD / ADAS Sensor map

But if we get into such detailed profiling for sonar sensors in the virtual environment, we will lose time and it will be expensive. Instead we can use a simplified profile that will serve the same purpose as a real sonar sensor and we can focus more on testing of the actual Device Under Testing (AD/AP/ADAS ECUs).

CHALLENGE #3: OBSTACLE SELECTION AND PRIORITIZATION

The model under discussion is generic and it should work in all scenarios. But for making it generic we need to address the following challenges

1. Identifying overall number of obstacles in the scenario
2. Identifying the number of obstacles within the range of each sensor
3. Identifing if the detected obstacles are really an obstacle (e.g. wheel stopper or a coin park flap in a parking slot is not an obstacle and the Sonar ECU should not stop parking when it detects it).
4. Identifing the obstacles from the final set of obstacles that requires further processing

CHALLENGE #4: OBSTACLE SIDE SELECTION

Once the obstacle list is freezed by the virtual sonar ECU, it has to then decide which side of each of the obstacles has to be processed (i.e. which side of the obstacle needs to be considered for checking for intersection with the sonar sensor profile). Multiple sides of the same obstacle may be visible at the same time to a sonar sensor. Please refer Fig 7 below which shows, in red, the possible sides that may be detected by the sonar sensor. But the question is, is it really necessary to consider all the sides for creating a Sonar ECU logic for a virtual validation setup?

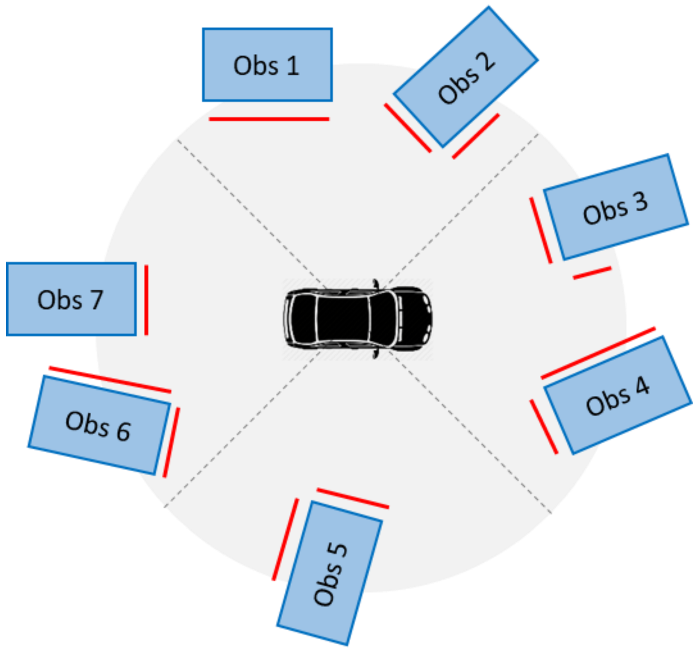
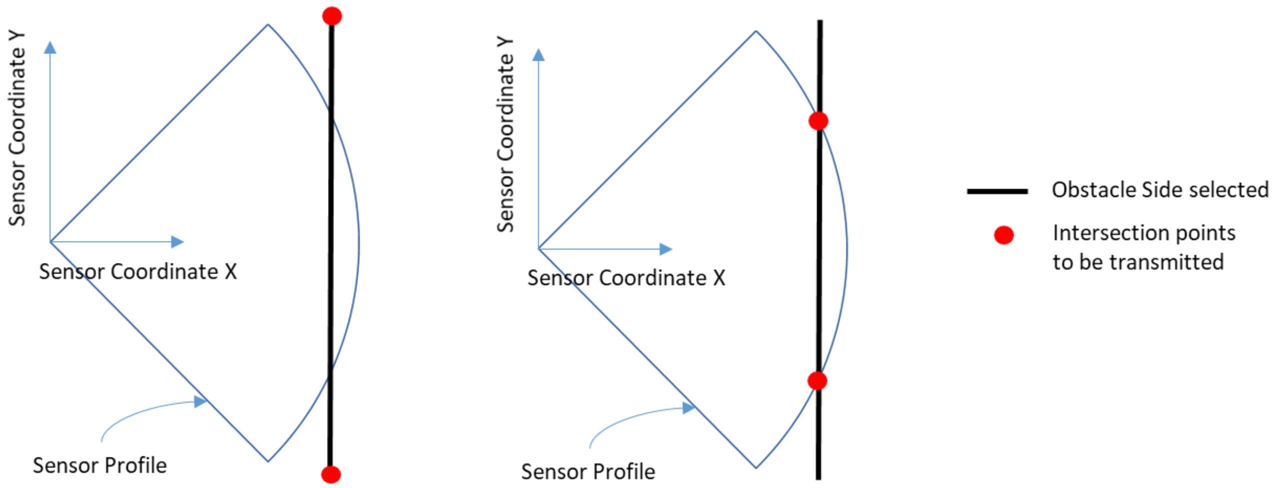


Figure 5: Possible sides detected by a sonar sensor for various obstacles around the Ego vehicle

CHALLENGE #5: COORDINATE DETERMINATION BASED ON INTERSECTION PROFILE

Once the side is selected we can send the respective coordinate information by two methods as shown in Fig 6 below. In Method 1, when the obstacle is detected, the entire selected side is transmitted to the DUT (AD/AP/ADAS ECUs) at once (Refer Method 1 in Fig 8 below). However, if there is a specific requirement that asks for gradual increment in the length of the selected side as it happens in the real world, then we may need to use specific logic to calculate the coordinates through some intersection logic applied to the profile of the sonar and the profile of the side under consideration (Refer Method 2 in Fig 6 below). The coordinates thus calculated by the intersection calculation will then be sent to the DUT (AD/AP/ADAS ECUs). But even if we consider simple profiles like a circle for sonar sensors and lines for obstacle sides, we still have a lot of configurations to think about.



Method 1
Full side transmission even if some portion of the selected side is in the Sonar sensor range

Method 2
Exact intersection points transmission

Fig. 8: Possible methods of Coordinate determination

CHALLENGE #6: PARKING SPACE ESTIMATION

In the case of simple sonar ECU logic, the obstacle is detected and the visible side of the obstacle is sent to the DUT (AD/AP/ADAS ECUs) and the story ends there. However, in the recent past, sonar ECUs have been loaded with additional tasks of determining if the space between detected obstacles is fit for parking or not. This functionality derived from the sonar ECUs is being used for features like Park Assist, Auto Park, etc. However, this task requires attention in the below areas:

1. How to calculate parking space if there is no obstacle or only 1 obstacle?
2. How to calculate parking space when there are two or more obstacles?
3. How to calculate parking space when two obstacles are at a certain angle to one another?
4. How to calculate parking space if there are two obstacles and a very long wall/hedge behind them
5. How to calculate parking space if there are multiple obstacles near the car out of which few are not in the range of the sensor but they will overlap with the estimated park spaces?
6. How to calculate parking space if the parking slot has wheel stopper/coin park flaps?
7. How to choose parking spaces if there are multiple parking spaces available?
8. How to prioritize the parking spaces for transmission to the DUT when multiple parking space is available?

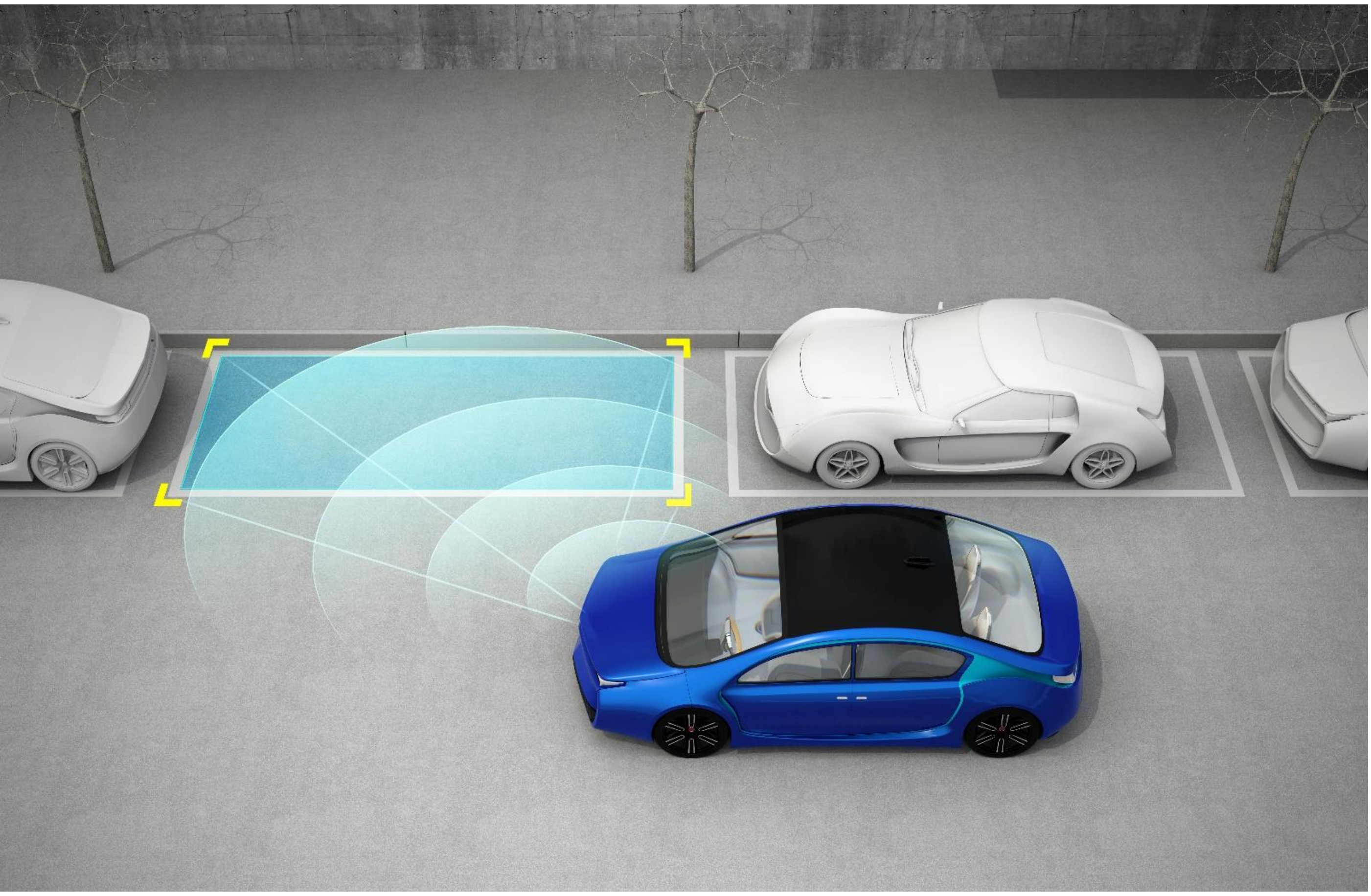
Hence, a generic logic for calculating parking space is required that caters to such varied requirements observed in various scenarios.

CHALLENGE #7: DATA TRANSMISSION CONSIDERATIONS

The data from sonar ECU may be sent to the DUT (AD/AP/ADAS ECUs) via hardwire or networks like CAN, etc. In either case, the rate of data transmission from sonar ECU to the DUT occurs at some defined intervals or events (like a regular cycle time of periodic messages in case of CAN). Also, there is a rate at which this sonar ECU model has to be updated. The model update rate should be quicker than the data transmission rate mentioned earlier, else there are chances of information loss. However, the update rate cannot be too fast and is limited by the following factors:

1. Model turnaround time already set in the simulation
2. Time is required to gather information from the surrounding and processing (Coordinate transformations, side selection, intersection calculations, etc.). Data transmission can start only after that. However, this should complete before the first data transmission to the DUT (say on CAN)
3. Type of Data / Information that has to be sent to the DUT
4. Size of Data/Information that has to be sent to the DUT
5. Transmission specifications provided by OEM / Supplier

It is very essential that we come up with the logic for the overall model to ensure that data transmission occurs properly as per the system specifications.



APPROACH TO OVERCOME THE CHALLENGES

Today, there are methods by which we can simulate Sonar sensors. Ready solutions are already available in tools like IPG Carmaker. However, Sonar ECU functional logics are not readily available as a generic solution in the market. Because of this, lab testing of AD/ ADAS ECUs becomes difficult as AD/ADAS ECUs require Sonar ECU input for creating maps using sensor fusion algorithms. Also, the record-replay method for Lab testing has its limitations in terms of the effort required and the variety of test cases that can be executed. So it is evident that the Sonar ECU model kind of approach is also needed in addition to the existing test methods to properly evaluate AD/ADAS systems in the Lab environment.

However, without resolving the challenges mentioned in the previous section, it will be difficult to proceed with this approach. Mentioned below are a few approaches and strategies that can be adopted to ensure a reliable Sonar ECU model.

1: APPROACH FOR HANDLING COORDINATES

Handling different coordinate systems within the same model may get a little confusing and inconvenient. In this case, the below approach may prove efficient:

Coordinate system	Coordinate handling approach
Sensor Coordinate system	Capture obstacle data like position, angle, height, etc. since we are interested in obstacles that are near to the sensor and their orientations with respect to the sensor
Global Coordinate	Creating obstacle maps, park space maps, obstacle co-ordinates calculations and park space co-ordinate calculations with respect to the overall scenarios including the Ego vehicle
OEM / Supplier Coordinate	To synchronize all the data received from different ECUs, OEMs sets it's own coordinate system for macroscopic calculations. Hence, the final data to be transmitted to the Autonomous ECU (via CAN, Flexray, other networks) can be expressed in this common coordinate system at the end of all calculations

2: APPROACH FOR SONAR SENSOR PROFILING

Instead of trying out very complicated patterns of actual sonar sensors, simplify by assuming some regular shape like a circle. Also, you may reduce the number of sensors mounted on the vehicle initially. E.g. (Refer Fig 9 below) you may start with two or four sensors and once the calculation model is at a matured stage, try adding more sensors for additional functionality or try complicating the sensor profile for more precision. Remember that this is a virtual environment and the obstacle data is already available with the simulation software unlike in real world scenarios. You can use this to your advantage by keeping your model simple and focusing more on the necessary functionality.

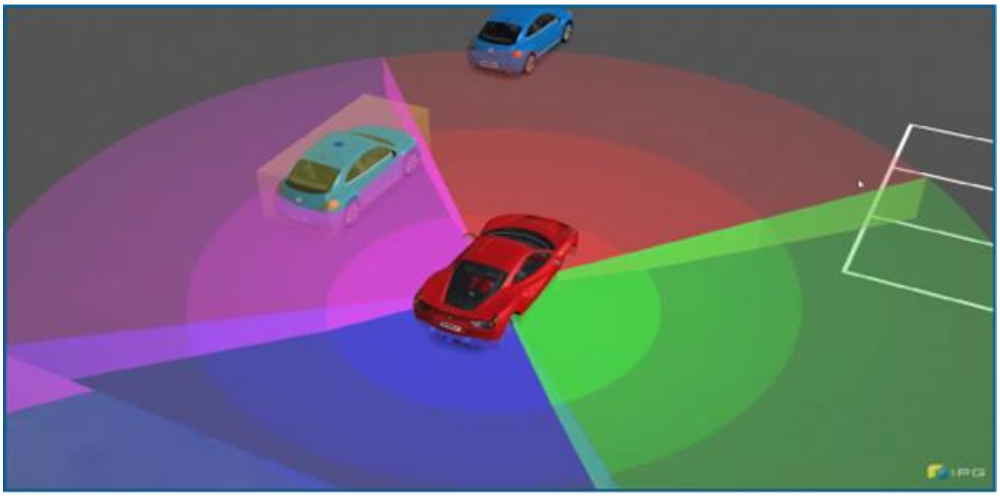


Fig. 9: Simplified Sonar sensor profile for the virtual environment (Image from IPG)

3: APPROACH FOR OBSTACLE SELECTION AND PRIORITIZATION

Since the scenarios are made using an environment modeling/simulation software tool, it is assumed that the obstacle count information (total and in each sensor view) can be directly obtained from this tool. However, for obstacle selection and prioritization the below factors needs to be considered:

1. Dimensions of the obstacles (Length, Breadth and Height) and it's spatial orientation in/near the ego car?
2. Is the obstacle fully or partially in sensor view?
3. Is the obstacle in the path that the ego car is traversing?
4. Is the obstacle near to the front end or rear end of the Ego car?
5. Distance between the obstacle and the Ego car's periphery?
6. Is the detected obstacle moving or stationary?
7. Total number of obstacles already stored for further processing?
8. Which obstacle is nearest to the front/rear bumper?
9. Which obstacle is nearest to the sides of the vehicle?
10. Is the obstacle in the parking slot or outside/around it?

Once such questions are answered, a decision model can be made to create a list of selected and prioritized obstacles. In case this decision is not proper, there are chances that Sonar ECU might miss detecting critical obstacles in the vicinity and send erroneous data to DUT (AD/AP/ADAS ECUs).

4: APPROACH FOR OBSTACLE SIDE SELECTION

To keep the processing simple for obstacle side selection, the transmission of a single edge of the obstacle is enough in a virtual environment. Please check Fig 10 below.

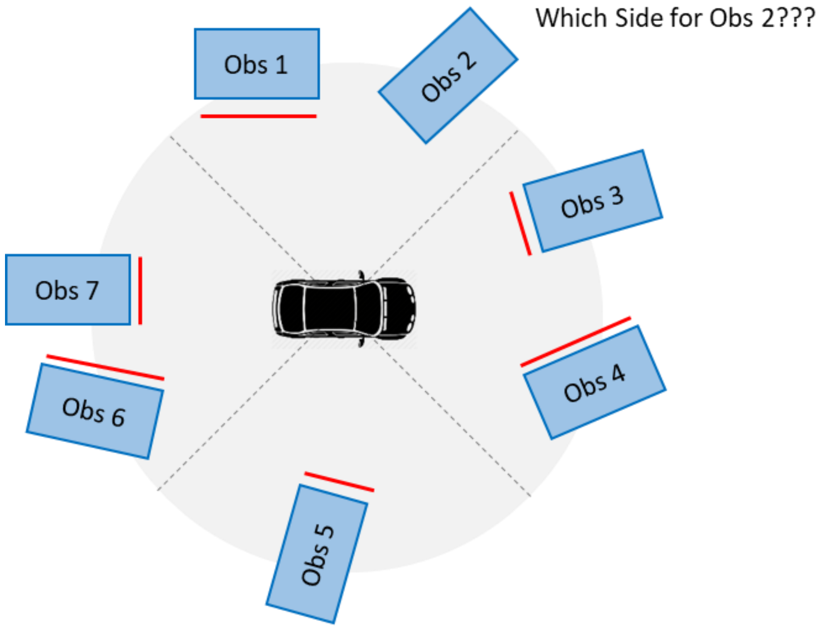


Fig. 10: Transmitting information of a single side of the obstacle to simplify the overall logic

Deciding which side to be selected will be based upon the below factors

1. Is the obstacle in the path that the ego car is traversing?
2. What is the angle of orientation of the obstacle w.r.t ego car (top view)?
3. Is the obstacle on the right or left side of the ego car?
4. Is the obstacle on the right or left side of the sensor?
5. Is it a continuous side/edge of the obstacle or just a sharp point?

For simplicity, we can also consider each obstacle as a 3D box/cuboid that encloses the obstacle completely. So usually, side selection happens for straight edges of this box. Otherwise, the shape/profile of the side also has to be taken into consideration.

5: APPROACH FOR COORDINATE DETERMINATION

Method 1(depicted in Fig 8) for coordinate transmission mentioned in the challenge in the previous section is very simple and it is sufficient to kick off the basic tests related to the DUT. However, if we go for Method 2 (depicted in Fig 8), a lot of intersection patterns need to be considered and programmed for. It is better to create all such possible patterns of Sonar Sensor profile and obstacle side intersections before creating this logic. A few samples are shown in Fig 11 below:

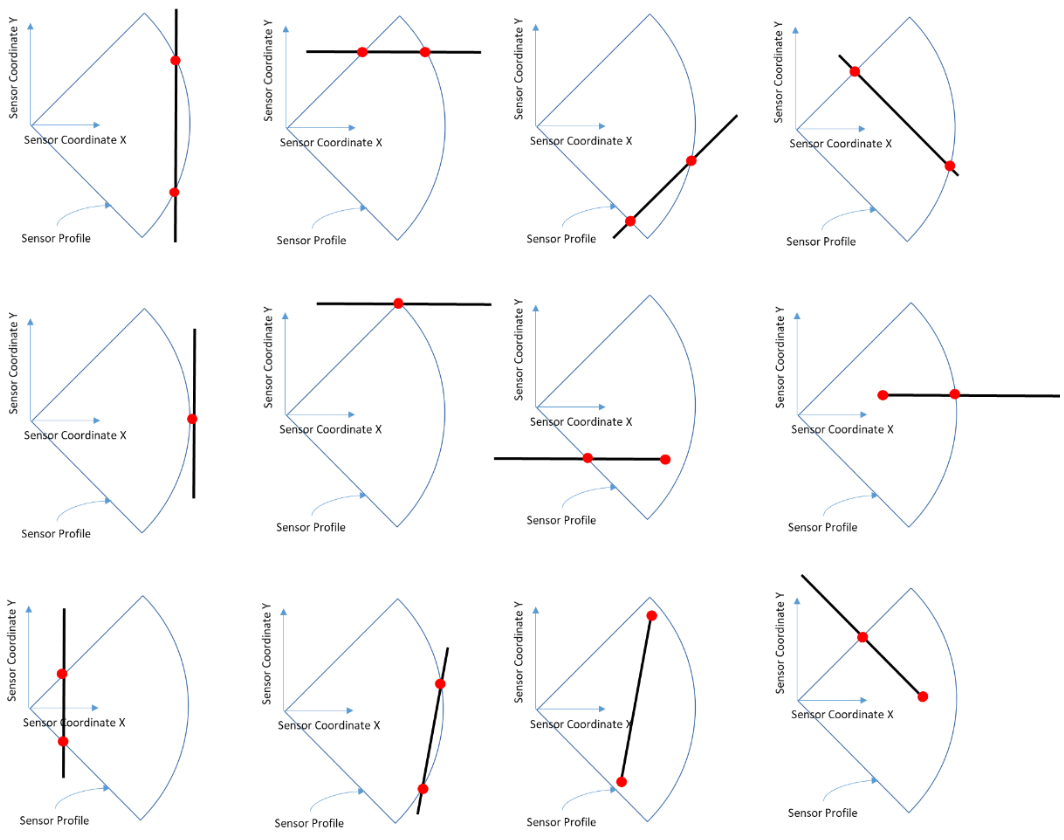


Fig. 11: Sample patterns showing the intersection of Sonar sensor profile with the obstacle side

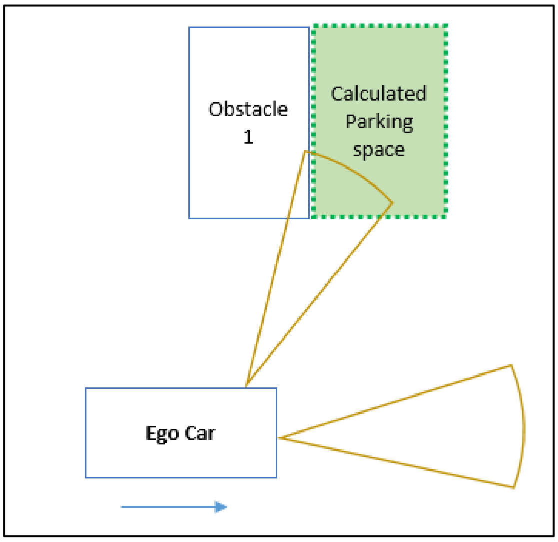
It is better to brainstorm all such possible cases and create a generic logic that will fit into any scenario.

6: APPROACH FOR PARKING SPACE ESTIMATION

For estimating the parking spaces, logic may be developed using the below approach:

1. Measure the "Gap" and "Depth" of the various spaces between multiple obstacles under consideration (obstacles may not be exactly parallel or perpendicular to each other)
2. In case of a single obstacle, decide upon a logic to estimate parking space and if that space will be parallel, perpendicular or angular
3. Select spaces that conform to the measurements for parallel, perpendicular or angular parking. This list should include parking spaces detected on the left as well as the right side of the ego car.
4. From the list of selected spaces, narrow down to one or two spaces depending on the proximity to the ego car and system requirements. Send the coordinates of this selected parking space to the DUT (AD/AP/ADAS ECUs)

It may happen in case of multiple obstacles that sonar detects only one obstacle due to its current position (refer Fig 12 below) and it may judge that there is a parking space near the detected obstacle. However, in reality, the estimated park space may be interfering with another obstacle in the scenario that is still not in the range of this sensor. Such inconsistencies have to be monitored very cautiously and have to be fixed to obtain a high fidelity model.



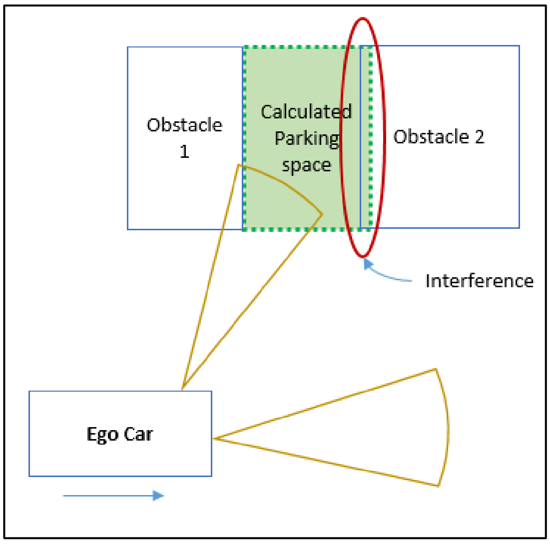


Fig. 12: (Left image) Single obstacle case, (Right image) Two obstacle case in which only one obstacle is detected, from Sensor point of view, both cases looks identical to it

7: APPROACH FOR DATA TRANSMISSION HANDLING

For preparing the data transmission logic, the below approach may be considered

1. Define an internal clock to keep track and sync of model data and the data transmitted over the network. Prepare the entire logic around this internal clock.
2. Define data cycle time to push newly calculated environment data to the DUT over the network (APAD/ADAS ECUs). This may be driven by the network requirements as defined/designed by the OEM/Tier 1.
3. At the start of each data cycle, extract obstacle data from the environment. Assuming 1ms for 1 obstacle data capture, you can retrieve data of 10 obstacles per sensor during the 1st 10ms of the data cycle (i.e. if you have four sensors, around 40 obstacle data may be captured during the first 10ms of the data cycle)
4. Monitor the processing time for the captured data and decide the earliest time at which data can be injected into the network. If processing time for captured data is large, then consider shortening the data capture time (e.g. from 10ms to 5ms in which case you can capture 5 obstacle data per sensor)
5. A rolling counter and checksum logic will prove to be very useful in keeping sync between the Sonar ECU logic model and the DUT

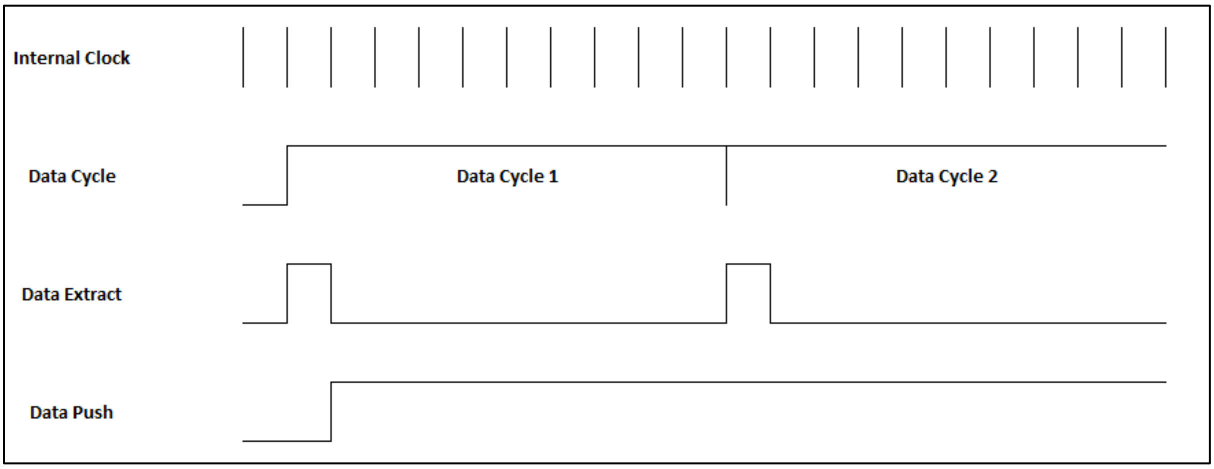


Fig. 13: Example of a Data handling and transmission model

CONCLUSION

In the coming years, virtual validation methods will become an integral part of almost all the OEMs and suppliers and there will be an increased demand for virtualization of ECUs that will enable quick and accurate testing in the lab environment itself. TATA ELXSI “SONAR ECU LOGIC” MODEL is one such step towards supporting the industry in this area. This model has already been tailored & tested in the “dSPACE HIL + IPG Carmaker” test environment and has been implemented in real projects.

The proposed solution is a generic Simulink model and has to be customized according to specific system requirements and software tool requirements. It may then be integrated into a larger Vehicle/Plant/Environment model and may be used for PC based MIL / SIL simulation or with RT HIL systems like dSPACE SCALEXIO.

These solutions take inputs from virtual sensors (e.g. object sensors, etc. in IPG Carmaker) of the environment simulation software tools (e.g. IPG Carmaker). It then processes the inputs and outputs high-level information like obstacle position, height and possible parking spaces to the AD/ADAS ECUs. Please refer to Fig. 14 below:

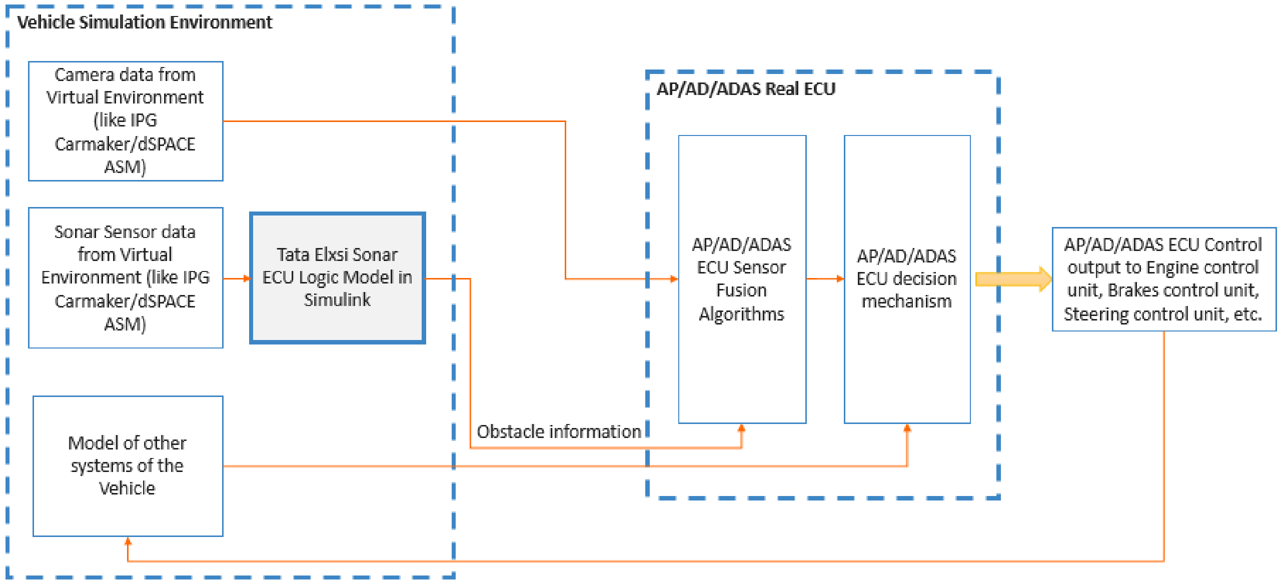


Fig. 14: Tata Elxsi Sonar ECU Logic Model based Test Ecosystem

Tata Elxsi can provide the generic model and required customization of this model as per specific customer requirements.

ABOUT TATA ELXSI

Tata Elxsi is amongst the world's leading providers of design and technology services for product design, engineering, solutions, electronics and software for the Automotive industry. Accredited with Automotive SPICE Level 5 certification, premium member of the AUTOSAR consortium and other industry partnerships help us gain a competitive advantage in the global market place.

Tata Elxsi is a leading provider of design and technology services for product engineering and validation across multiple industry verticals such as Broadcast, Communications, Healthcare, Transportation. Transportation business unit provides services in the domain of automobiles that range from software-based services like software design, development, and testing as well as Hardware-based services like product design development and evaluation. The transportation business unit has a wide range of experience that varies from classical control units like engine, body, and steering to futuristic technologies such as Autonomous features, AI and IoT integration in automobiles.

Tata Elxsi has supported multiple OEMs and Tier-1 suppliers globally for testing and validation of their programs. Tata Elxsi has experience in conducting software in loop (SIL), Model in loop (MIL) and Hardware in loop (HIL) testing for various automotive programs. Designed and developed 40+ HILS for testing single and multiple ECU's. (Infotainment, BCM, Chassis, and Powertrain ECUs).

REFERENCES

1. Rethinking Car software and electronics architecture, McKinsey
2. Autonomous Vehicle market, Allied Market Research, 2018
3. Global Autonomous Driving Market Outlook 2018, Frost & Sullivan
4. The Future of Autonomous Driving, Bain & Company
5. Challenges in Autonomous Vehicle Testing & validation, ResearchGate publications
6. Area-wide real-world test scenarios of poor visibility for safe development of the automated vehicle, European Transport Research Review
7. Future certification of Automated/Autonomous Driving, unece.org
8. How much testing will prove automated cars safe, Automotive-IQ
9. Autonomous Driving Main Technology Challenges - ALTRAN
10. Testing of Autonomous Vehicle – Department of Motor Vehicle, California.Gov
11. Waymo Safety report, 2018
12. Global ADAS Ultrasonic wave component market size, Statista
13. Market forecast for connected and autonomous vehicle, Transport systems Catapult
14. Global Autonomous Vehicle sensor market, Market Research Future
15. Global Automotive LiDAR Market, Globenewswire
16. Monetizing the rise of Autonomous Vehicles, Goldman Sachs
17. Ultrasonic sensor product detail, muRata

