JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Adaptive Cruise Control with Sensor fusion

[1]Dr. Pramod Kumar Naik, [2]Keerthi Reddy V, [3]Juvvi Arun Sai, [4]Jatin Pabothu, [5]Harshitha Reddy S
Associate Professor, Students
Computer Science and Engineering,
Dayananda Sagar University, Bangalore, India.

Abstract: *According to the research and surveys conducted Higher levels of accidents occur because of slip-ups or carelessness of Drivers. More elevated levels of setbacks happen due to the mistakes or inconsiderateness of Drivers. Very few of the many purposes behind this need are inexperienced drivers, improper executions of driver help systems. The need for a good driving assistance system is high. Hence, we are developing and implementing an advanced system with "Adaptive Cruise Control with Sensor Fusion" in our project.*

This project explores the ACC response of different sensor fusion configurations and how they interact in a virtual simulated curved road environment with other vehicles. Driving in heavy traffic or keeping a safe distance to the preceding vehicle calls for a high level of concentration. The adaptive cruise control (ACC) can reduce stress for the driver by automatically controlling vehicle speed and maintaining a predefined minimum distance to the preceding vehicle. But 1 Challenge that comes with these benefits is accuracy and its implementation in curved roads. The proposed system presents the solution for this problem. Scenarios and 3D animated videos are created with cars and lanes in a controllable environment in MATLAB using Simulink 3D animation. The ego car detects the other cars and accelerates and deaccelerates the speed accordingly to avoid collision. The ego car also changes the lanes and overtakes the other car to avoid stopping of the cars abruptly when other cars slow down, together this creates a safe driving experience. Two different toolboxes are implemented in different vehicles i.e., Model predictive control (MPC) and Proportional integral derivative (PID) which takes different inputs compared to one another.

***Index Terms* -** ACC (Adaptive Cruise Control), ADAS (Advance Driving Assistance System), CACC (Conventional Adaptive Cruise Control), (MPC) Model predictive control and (PID) Proportional integral derivative

I. INTRODUCTION

Advanced Driving and System requires sufficient input for performance optimization to reduce road accidents and losses. Adaptive cruise control is an advanced version of cruise control, the advanced driving-assistance system for road vehicles that automatically adjusts the vehicle speed to maintain a safe distance from vehicles ahead. Adaptive Cruise Control keeps a set haste and distance to avoid collision with the vehicle in front. Sensor fusion technology plays a pivotal part in the Advanced Driving Assistance System. It combines data from multiple sensors similar as the camera, radar, and lidar to form a detailed image of the driving terrain. An adaptive cruise control system is a control system that changes the velocity of the ego vehicle in response to challenges on the road. As in regular cruise control, the driver sets the asked speed for the car. The goal of ACC is to avoid inordinate speeding and short-advance- related accidents. There are numerous driving scenario's that comprise accident hot spots, such as intersections and twisted roads. The usefulness of a simulated driving terrain allows experimenters to collect, ameliorate, and tinker with sensor fusion modalities to apply the most effective ADAS. The success of a robust and dependable sensor fusion safety system is dependent on processing power, cost, and fast objectification into the consumer request. Data fusion snappily becomes a complicated procedure when considering all the content areas for automotive safety operations. The initial ACC sensor layout gradually builds upon it until reaching the final sensor layout design best suited for a curved road scenario with flowing traffic. The driving scenario is recreated in MATLAB's Simulink using MathWorks based code with some adjusted Simulink blocks.

For the ACC to perform correctly, the ego vehicle must judge how the lane in front of it curves, and which car is the lead car', that is, in front of the ego vehicle in the lane. A standard scenario from the perspective of the ego vehicle is shown in the following figure. The ego vehicle (blue) travels along a curvy road. At the start, the lead car is the pink car. Also, the purple motor vehicle cuts into the lane of the ego vehicle and becomes the super-eminent vehicle. After a while, the purple auto changes to another lane, and the pink vehicle becomes the super-eminent motor vehicle again. The pink motor vehicle remains the super-eminent motor vehicle subsequently. The ACC system must respond to the revision in the super-eminent auto on the road. Usual ACC schemes depend substantially on the range and range rate measures attained from radar and are aimed to perform along straight roads. An instance of such a system is given in Adaptive Cruise Control System Using Model Predictive Control (Model Predictive Control Toolbox) and in Automotive Adaptive Cruise Control utilizing FMCW Technology (Radar Toolbox).

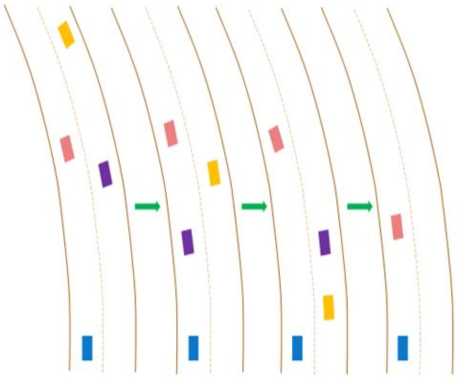

Fig.1.1 An Overview of Driving Scenario

Relocating from advanced driver-assistance system (ADAS) designs to more autonomous systems, the ACC must handle the following challenges evaluating the comparative positions and speeds of the cars that are near the ego vehicle and that have significant side motion compared to the ego vehicle.

Estimating the lane ahead of the ego vehicle to detect which car in front of the ego vehicle is the closest one in the same lane. responding to aggressive maneuvers by other vehicles in the surroundings, in particular, when another vehicle cuts into the ego vehicle lane. This exemplification demonstrates two main extensions to subsisting ACC designs that meet these challenges adding a sensor fusion system and updating the controller design based on model predictive control (MPC). A tracking system and sensor fusion that uses both visual and radar sensors provide the following advantages. It combines the better lateral measurement of position and speed obtained from vision sensors with the range and range rate measurement from radar sensors. A vision sensor can determine lanes, deliver an estimate of the lateral position of the lane compared to the ego vehicle, and position the other cars in the setting comparative to the ego vehicle lane. This example assumes ideal lane detection. An improved MPC controller adds the ability to respond to more aggressive maneuvers by other vehicles in the environment. In contrast to a conventional controller that uses a PID design with unchanging gains, the MPC controller regulates the speed of the ego vehicle while maintaining a strict safe distance constraint. Therefore, the controller can apply more aggressive maneuvers when the environment changes quickly in a comparable to what a human driver would do.

II. OBJECTIVE

The primary objective of a project on adaptive cruise control using sensor fusion is to develop a system that can adjust the speed of a vehicle based on the traffic conditions and maintain a safe distance from other vehicles on the road ensuring safety.
Ensure safety: The second objective is to ensure the safety of the vehicle and the passengers by avoiding collisions with other vehicles and maintaining a safe distance from other vehicles
Implement adaptive cruise control: The third objective is to implement an adaptive cruise control model that can adjust the speed of the vehicle based on the distance from the vehicle ahead, the speed of the vehicle ahead, and the relative velocity between the two vehicles.
Test and optimize: The fourth objective is to test and optimize the system to ensure its reliability and effectiveness in various driving conditions and scenarios.

III. LITERATURE SURVEY

In this paper they present a novel attack-resilient sensor fusion method for vehicle platooning, using spatial information exchanged in cooperative adaptive cruise control. The Experimental results via PreScan simulation show the new algorithm outperforms traditional sensor fusion methods, in both maintaining the positional error in a smaller range and suppressing the uncertainty of the fusion result. And also, they have compared the performance of new sensor fusion method with existing algorithms when the vehicle platoon is under different attacks and controlled by different CACC control methods. [1]

This paper presents the design, development, implementation and testing of an enhancement to commercially available ACC systems, based on introducing vehicle-to vehicle communications, to produce CACC. The design has been enforced in four production Infiniti M56s vehicles equipped with DSRC devices for information exchange among automobiles. The CACC controller system benefits off of wireless communication information, introducing feed forward terms in the control logic, to enable significant reductions in intravehicular gaps. The CACC clearly showed improvements in response time and string stability, denoting the eventuality for a CACC system to attenuate disturbances and ameliorate freeway volume. [2]

This paper presents an approach for detecting target vehicle cut- sways for ACC systems. Input from colorful detectors (radar, camera, and lidar) are combined to estimate target vehicles' positions in their lanes. A literacy-grounded lane change intention estimation algorithm identifies target vehicles that may change lanes into the path of the pride vehicle. A control strategy grounded on robust MPC is presented which considers safety concerning all applicable target vehicles without the need for heuristics to elect a single primary target. The proposed methodology is shown to ameliorate the performance of the ACC system as compared to a being design, grounded on colorful units that consider amenities and safety. [3]

This paper, focuses on the practical perpetration of CACC, using a test set- up conforming of six passenger vehicles and provides a short overview of string stability generalities. String stability is an essential demand for the design of vehicle-following control systems that aim for short-distance following. A drastic increase in road outturn and, in the case of large vehicles similar to exchanges, etc., a drop in energy consumption and emigrations can be anticipated. Current advancements are concentrating on the perpetration of features similar to fail safety. Driving at slight time breaks requires a largely trustworthy CACC system and, if it fails, will do so gracefully. [4]

A system for mongrel adaptive voyage control (HACC) on high-speed roads designed as a combination of a radar-grounded ACC and visual perception is presented. The combination of radar and vision leads to a system with enhanced performance, able of handling several tasks concertedly using a common knowledge base. The advantages of a combination of the two different types of detectors are bandied. A description of the visual lane discovery and shadowing procedure is given, followed by an overview of the vehicle discovery, thesis generation, and tracking procedure. The system can handpick an applicable position of performance depending on the attack status or the performance of the experts.[5]

IV. METHOD

4.1 VEHICLE MODELS

4.1.1 VEHICLE ONE

- For both the ego vehicle and the lead vehicle, the dynamics between acceleration and velocity are modelled as:

$$G = \frac{1}{s(0.5s+1)}$$

- The main scenario of the lead car (Vehicle 1) is to take the initial reference speed and convert it to speed and position output.
- Here, the transfer function is used for Acc Dynamics to convert acceleration to speed. Set the denominator coefficient to (0.5, 1, 0).
- We need to separate integrator from transfer function, this changes the step response of the system which we use the integrator to set the initial speed for our vehicle model.
- We can make the vehicle follow the reference speed input.
- The vehicle starts with an initial speed of 10m/s and speeds up to 20m/s.
- Vehicle have dynamic limitations for the accelerations and deceleration. Therefore, to simulate their behaviour more realistically we use a saturation block for the acceleration command.
- We use another saturation block to make sure that the vehicle does not have negative speed output.

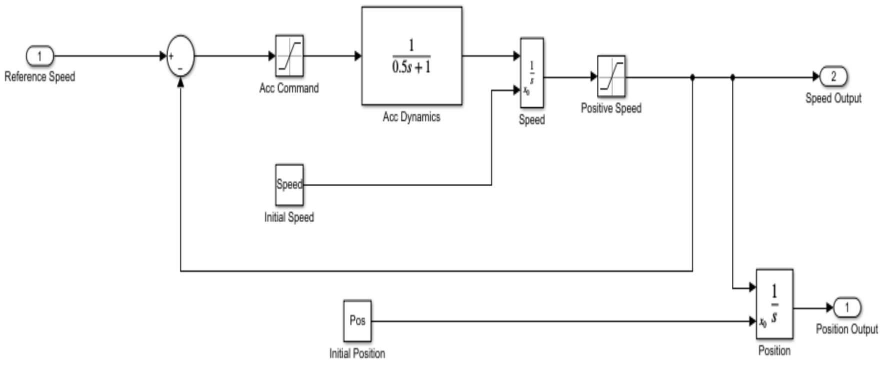

Fig 4.1 Vehicle 1 Subsystem

- Now, we need position output of the vehicle. To obtain the position we will integrate the actual speed of the vehicle. The vehicle starts and from the initial position 100m and goes on Constant speed.

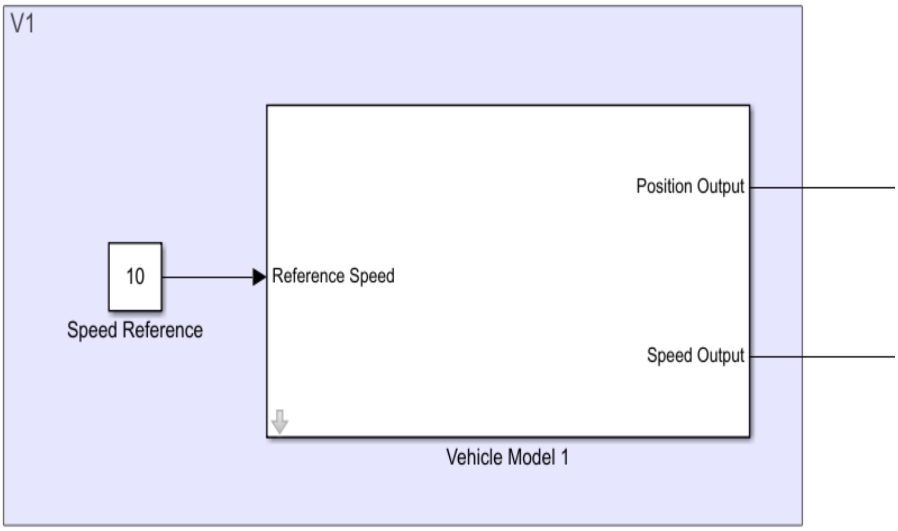

Fig 4.2. Vehicle 1 model

4.1.2 VEHICLE TWO

- The second vehicle model is mirrored after the first vehicle with same inputs and outputs however we need to set the speed relative to the first vehicle and make sure to set the parameters so that the relative distance does not become zero which would result in a collision
- We add a MPC controller to the second vehicle. The standard MPC controller requires 5 outputs which are:
- Set velocity
- Time gap
- Longitudinal velocity
- Relative distance
- Relative velocity

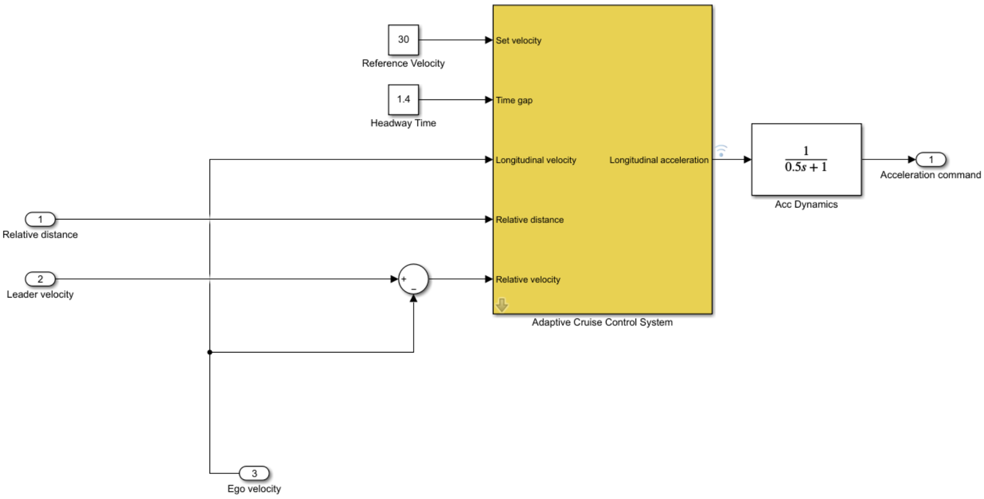

Fig 4.3. MPC vehicle following mode

To make vehicle two successfully follow lead car we need to first provide the above given parameters and then set the initial speed and the position of the vehicle with respect to vehicle 1

Next, we need to tune our MPC model as at this point it is inefficient and time consuming this is d1 by changing the control behaviour to make it more robust and increase the sample time all the while decreasing the prediction steps this will significantly reduce computational cost.

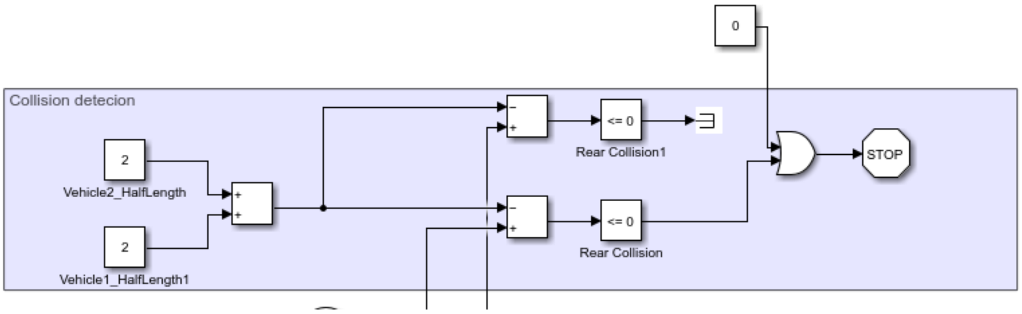

Fig 4.4. Collision Detection

- Running the simulation, from fig. 6.4 we find out that the relative distance between the vehicles at one-point falls to below zero but the simulation keeps running as the distance becomes negative soon after to avoid such situations, we can stop the simulation when a collision occurs by using a stop block the condition of this block can be "when the relative distance is >= 0".After which it stops immediately after the start of the simulation to avoid this.
- Next, as seen in fig 6.5 we add a lateral motion model to influence the vehicle laterally. The lateral motion models we use is simple angle model after which we need to update the animation by adding relevant inputs to see implementation.
- The angle can only be generated when the vehicle is turned around in Y axis so we need an array for inputs therefore we create a bus. In this bus we connect the vehicle lateral motion.
- Now we integrate longitudinal motion and lateral motion of the following vehicle which is d1 by defining initial lateral angle then the initial lateral and longitudinal position and integrate speed of the vehicle according to the angles, therefore for our lateral motion we must add the signal which comes from vehicles lateral displacement .

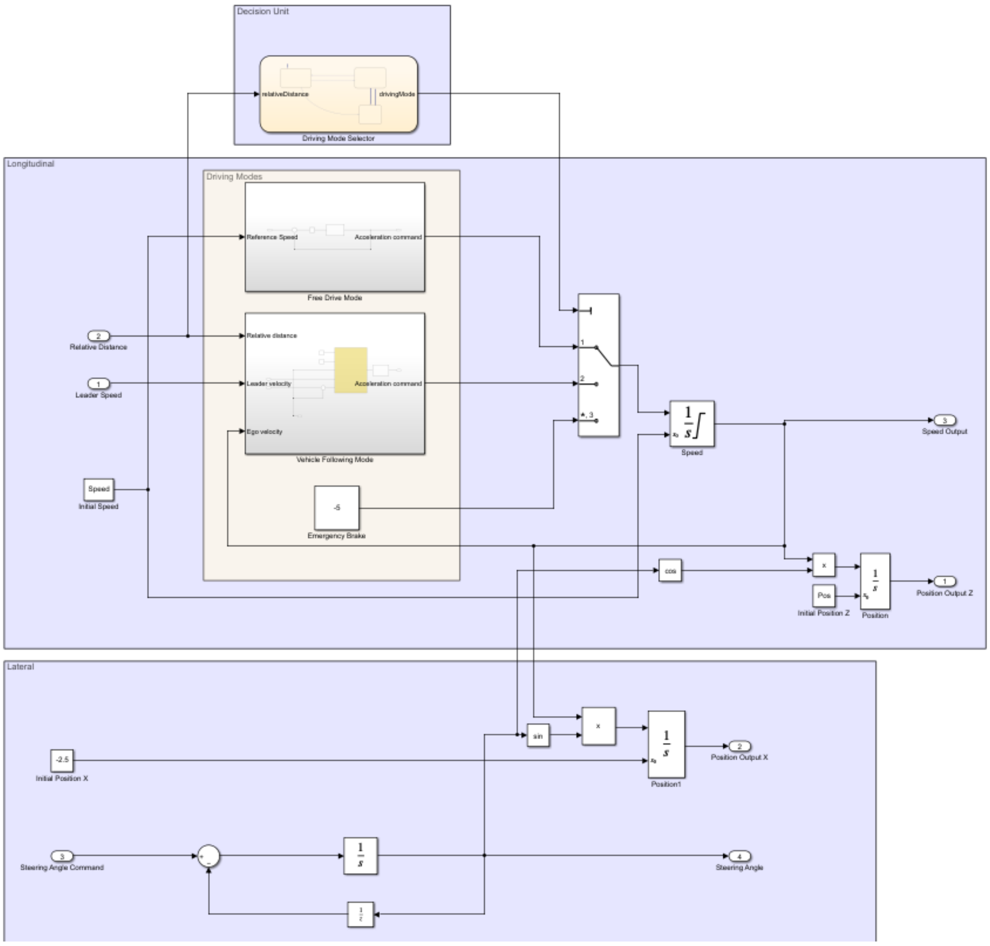

Fig 4.5. Longitudinal and lateral view subsystems

- Now we implement driving modes for the vehicles to drive freely when the distance between the vehicles are far more than constant value for which we make a subsystem of our adaptive cruise control and set relevant inputs outputs to be consistent. The output of the previously engineered adaptive cruise control should be acceleration and the parallel mode we are going to set is also going to provide us with acceleration. The activation of this parallel mode is d1 when the distance between the vehicles is far more than the constant or safe value. This is implemented when there is no vehicle ahead, it makes it so that the vehicle can drive freely on a highway.

- The acceleration generation of this subsystem driving mode block will be very similar to the lead vehicle which is already in a free driving mode. Ref fig. 6.1. Finally, we use a switch to simply switch between the 2-driving mode present which are free driving mode and adaptive cruise control. This switch will generate value 1 if input is smaller or equal to 30 the vehicle will switch to second driving mode which is adaptive cruise control but if the distance is more than 30 the vehicle switches free drive mode using the switch. The default state of the driving mode will be free drive then depending on the distance will switch into the adaptive cruise control mode.
- A third subsystem which is the emergency brake system Fig. 6.6 which applies immediate and powerful brake compared to adaptive cruise control mode. Implementing direct state after which the following vehicle switches back and forth, when the vehicle closes in to the leading car it switches from transition from initial state or also known as free drive mode to emergency brake state without spending sample time in vehicle following mode or adaptive cruise control mode. We then extend the switch model to accommodate the 3rd driving mode state to brake state after which the distance is increased between the following and the lead car which then triggers the switch to change the mode to free driving which then again changes to following mode.

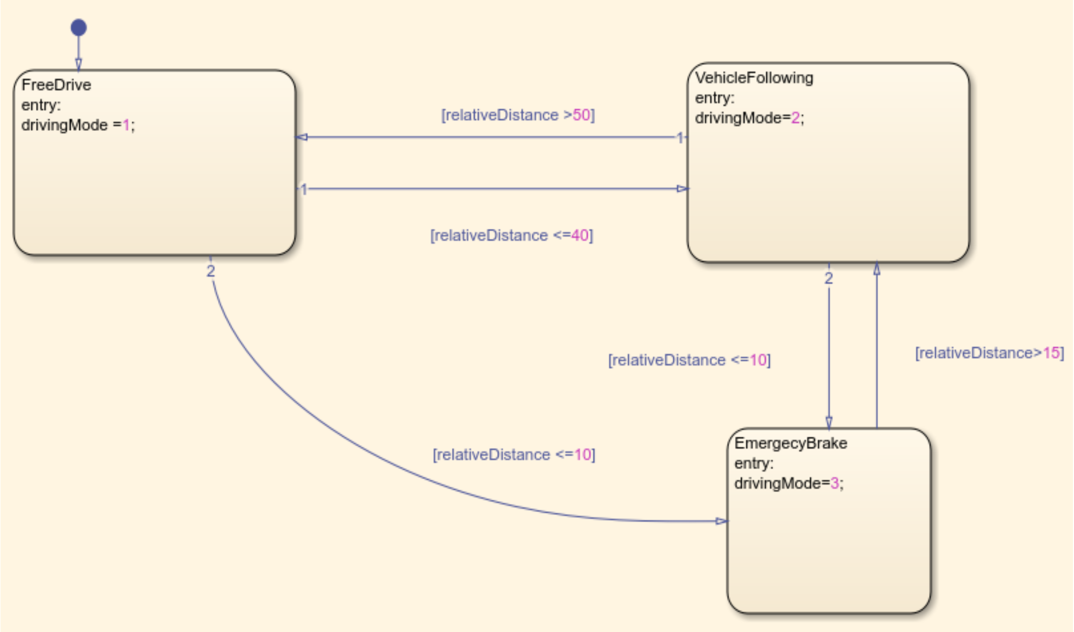

Fig 4.6. Emergency Breaking System

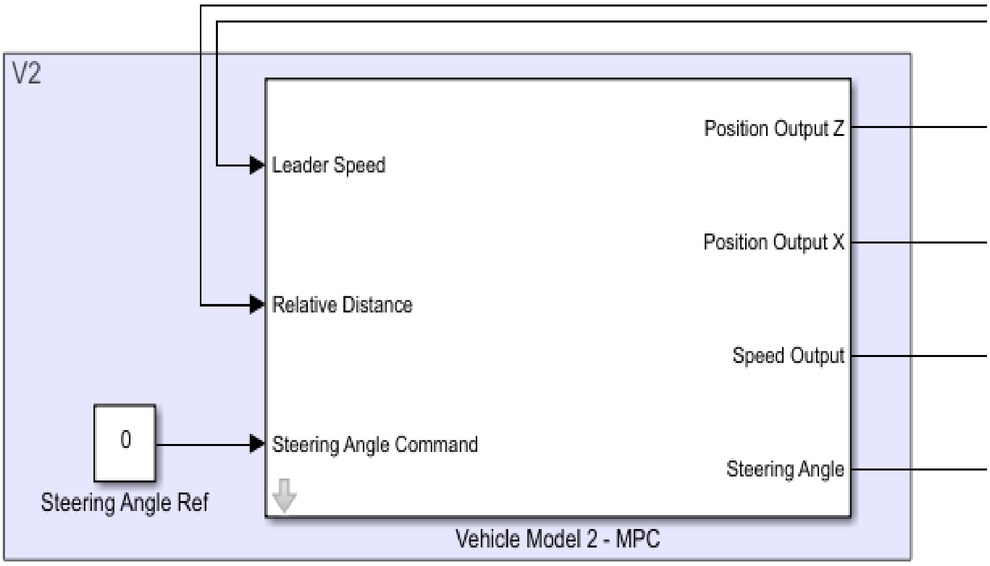

Fig 4.7. Vehicle 2 model

4.1.3 VEHICLE THREE

- Implementing the third vehicle using PID controller instead of MPC. Third vehicle follows the second.
- PID control box also known as proportional integral derivative requires the following inputs:
- Default spacing
- Time headway
- Relative distance
- Relative speed

- Ego vehicle speed

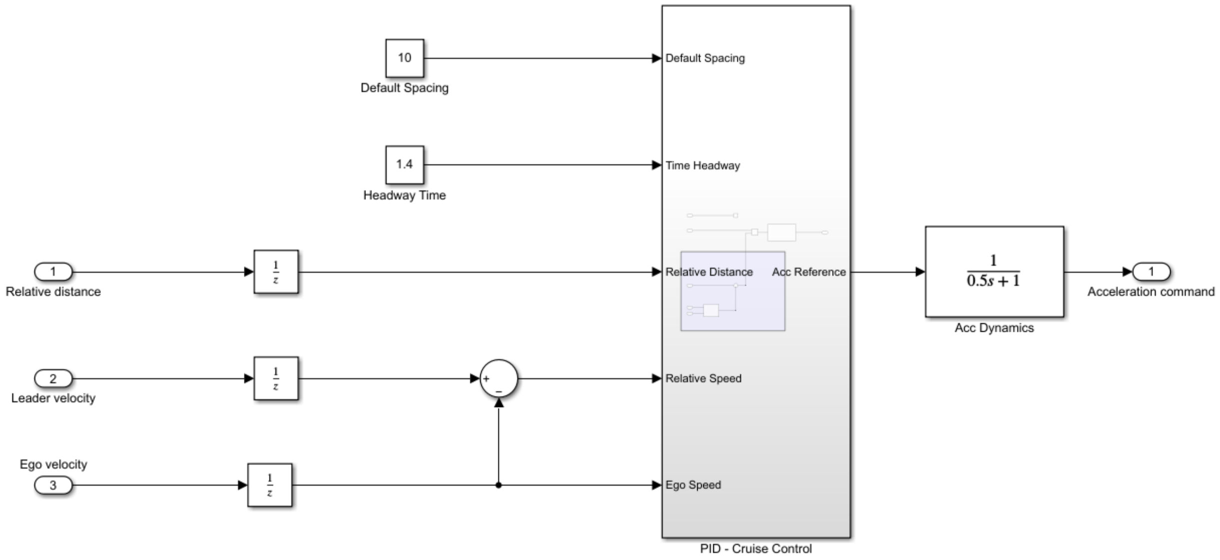

Fig 4.8. PID vehicle following mode

- This outputs the relative acceleration. The reference distance is calculated by the following formula: **ego vehicle speed *time headway + default spacing = reference distance.**

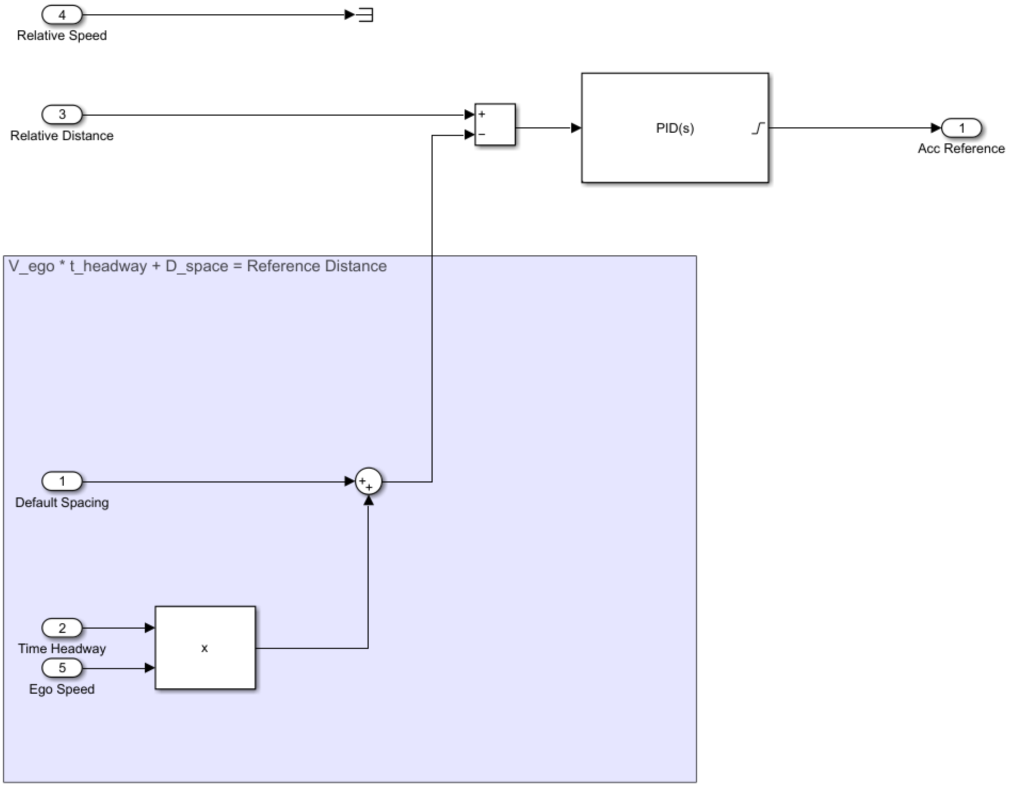

Fig 4.9. Free Drive Mode

- After configuring the third vehicle we map it to the visualization block which has similar outputs like vehicle model 1 and vehicle model two
- The collisions are calculated using longitudinal scenarios. As shown in fig. 6.5
- In addition to the existing parameters, we add an additional feature called lane changer. The previous vehicles had a steering angle of 0 which prohibits them from changing lanes. The lane changing is d1 in this vehicle by changing the steering angle which has a maximum and minimum angle of 0.2 and -0.2
- We input the steering angle to the vehicle model 3 for lane changing. A dashboard scope is created to observe the changes in the lane.

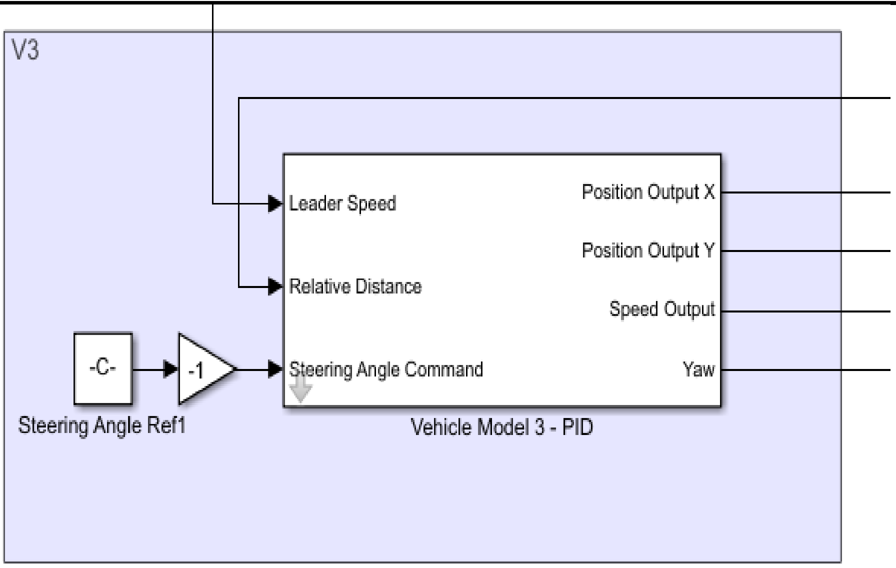

Fig 4.10. Vehicle two model

4.2 VISUALIZATION

- Take the inputs for x, y and z and vectorise the poles of the vehicles which includes x y and the yaw angle.
- The inputs x, y and yaw angles of each vehicle are mapped to the visualization block.
- For the leading vehicle we are not controlling the lateral movement of this vehicle this is basically moving on a predefining speed and direction and hence the x, y and yaw angle would be zero. Minus means we go from the lower lane and go right
- If the centre points do not overlap the collision does not happen

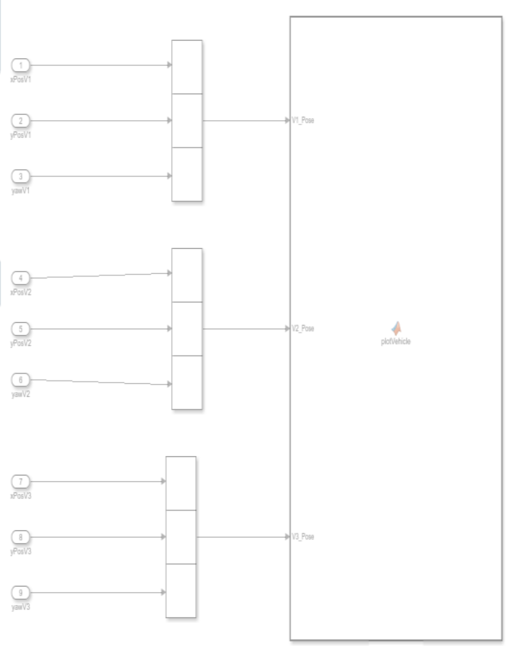

Fig 4.11. Car positions mapped to Visualization block

We can visualize the simulation by Adding 3D animation block. We need to specify the source file for the animation block which allows us to specify the 3D objects and their parameters. The only parameter we need to change during the simulation is a translation vector of the vehicle. In this stage we only simulate the longitudinal motion. We avoid collision by setting the initial positions and the initial speed of the vehicles.

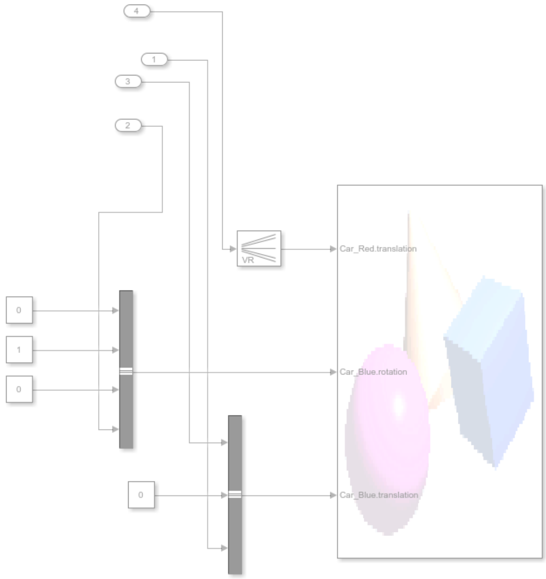

Fig 4.12. 3D Visualization Block

4.3. UNREAL ENGINE

- Integrating unreal engine to simulate and visualise the driving scenarios on Simulink.
- Importing a block 3D vehicle with ground truth simulation to the model in which the inputs are x, y and yaw angle. These contains different colours, car type, initial position and initial rotation, name of the vehicle and the sample type and the yaw angle is constant. Here we import three different types of vehicles.

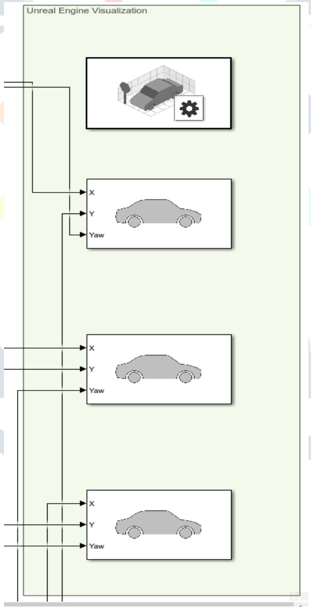

Fig 4. 13 Unreal engine Visualization

V. RESULTS AND DISCUSSION

5.1 VEHICLE ONE

For vehicle one(red), the lead car, there is no Adaptive cruise control applied, it drives in a free driving mode throughout the simulation.

Fig 5.1. Vehicle One

5.2 VEHICLE TWO

For vehicle two (orange), ACC is applied with the use of MPC toolbox. It accelerates, de-accelerates its speed according to the lead vehicle. Applies emergency breaking if the distance between them is found to be less.

Fig 5.2. Vehicle Two

5.3 VEHICLE THREE

For vehicle three (white), ACC is applied with the use of PID toolbox. It accelerates, de-accelerates its speed according to the lead vehicle. Applies emergency breaking if the distance between them is found to be less.

Additional feature of the vehicle three is that, it overtakes the vehicle in front to avoid collision when an acceleration is detected

Fig 5.3. Vehicle Three

5.4 THE SIMULINK

All the cars, subsystems put together forms the whole model of Adaptive Cruise Control.

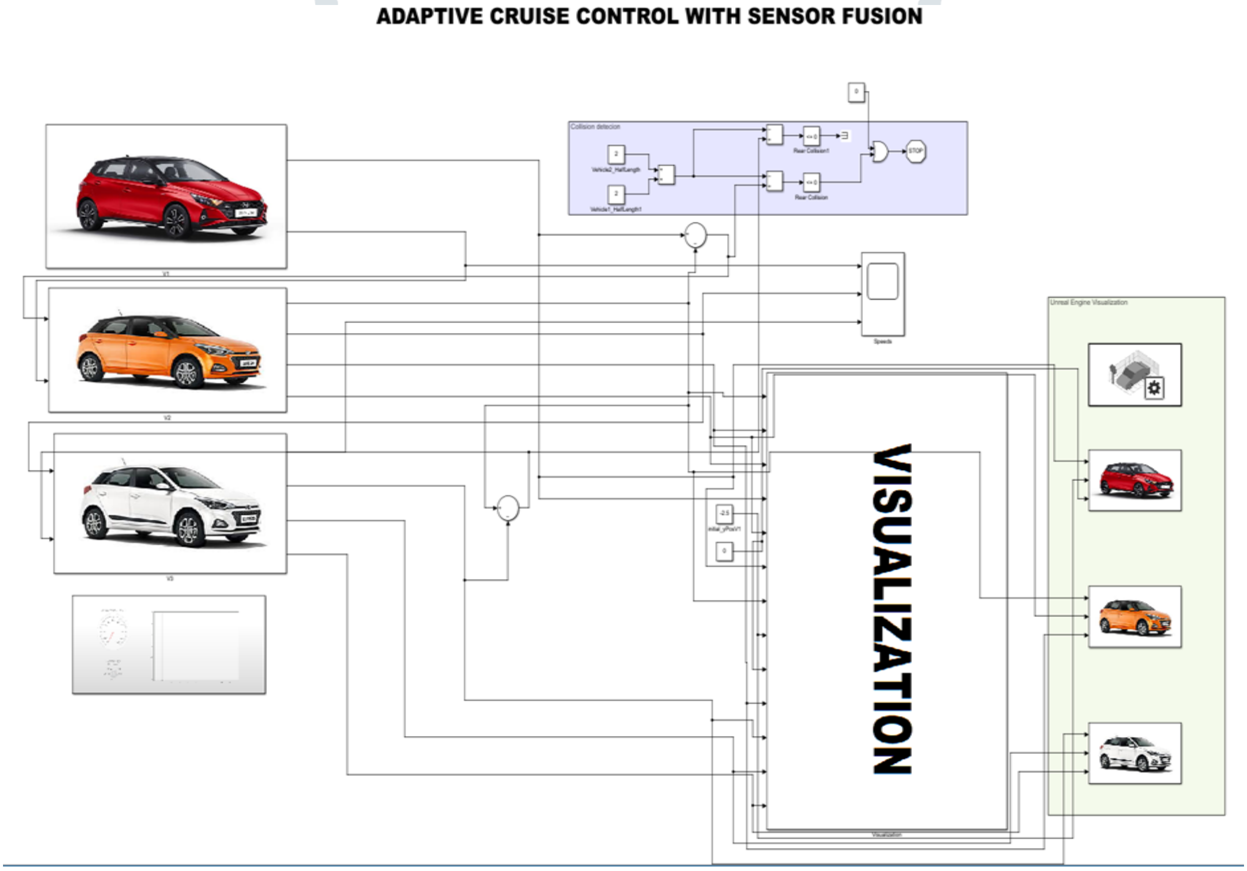

Fig 5.4. The ACC Model

5.5 SPEED

The speeds of the three vehicles throughout the simulation.

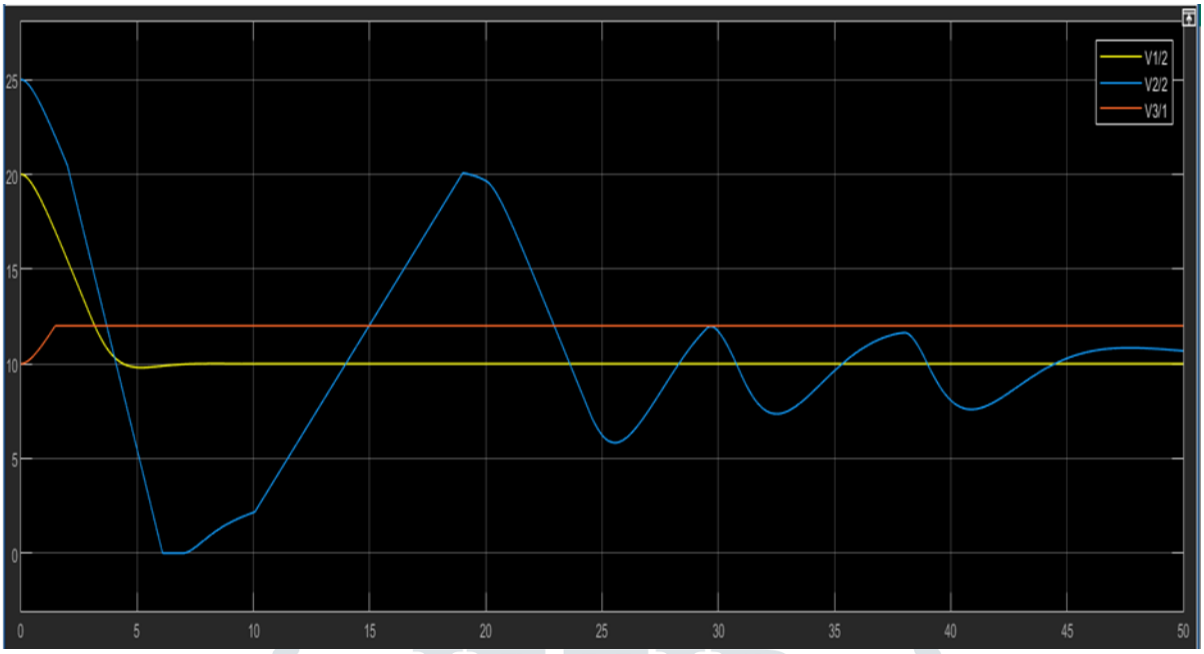

Fig 5.5. Speed of the vehicles

5.6 SEQUENTIAL FORM

A Sequential representation of the driving modes of the Adaptive Cruise Control applied Cars.

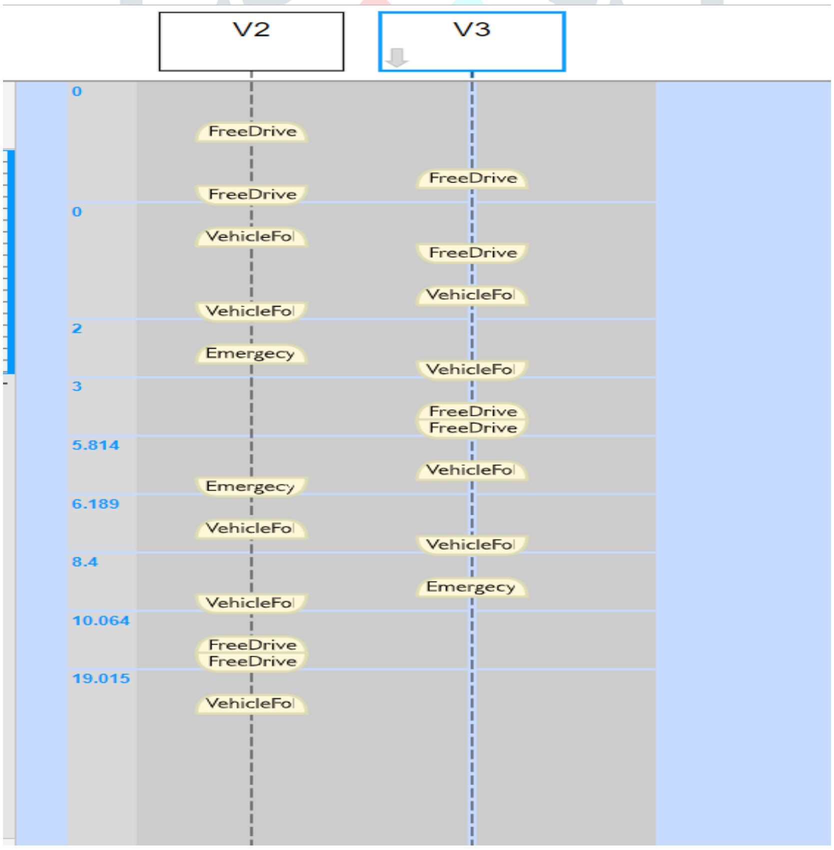

Fig 5.6. Sequential form

VI. CONCLUSION

In this project we have implemented an integrated Adaptive Cruise Controller by fusing some sensors such as LiDAR's and cameras by fusing them together. We have designed three vehicles in the model out of which two vehicles shows the difference between the adaptive cruise control vehicle and the normal vehicle using few control tool boxes like Model Predictive Controller (MPC) and Proportional Integral Derivative (PID) in which accelerates and deaccelerates its speed according to the neighboring vehicles and also overtake the car ahead to avoid collision, also using the longitudinal and lateral view of the vehicles. We test it in Simulink using software co-design where we have created the subsystems in the Simulink and mapped it to the 3D visualization block where the animations are created.

VII. REFERENCES

[1] Pengyuan Lu, Limin Zhang, B. Brian Park, Lu Feng, “Attack-Resilient Sensor Fusion for Cooperative Adaptive Cruise Control”, International Conference on Intelligent Transportation Systems (ITSC)/2018.

[2] Vicente Milanes, Steven E. Shladover, John Spring, Christopher Nowakowski, Hiroshi Kawazoe, Masahide Nakamura, “Cooperative Adaptive Cruise Control in Real Traffic Situations”,International Journal Science and Research(IJSR)/ 2020.

[3] Ashwin Carvalho, Alek Williams, Stephanie Lefebvre & Francesco Borrelli, “Autonomous cruise control with cut-in target vehicle detection”,International Journal of Digital Application & Contemporary research/2018.

[4] Jeroen Ploeg, Bart T. M. Scheepers, Ellen van Nunen, Nathan van de Wouw, Henk Nijmeijer, “Design and experimental evaluation of cooperative adaptive cruise control”, International IEEE Conference on Intelligent Transportation Systems (ITSC)/2019.

[5] U. Hofmann, A. Rieder, E. D. Dickmanns, “Radar and vision data fusion for hybrid adaptive cruise control on highways”,International Journal Science and Research(IJSR)/ 2020.

[6] Subhash Satsangi, Prashant Panse, “Adaptive Cruise Control using Fuzzy Logic”, International Journal of Digital Application & Contemporary research/2020.

[7] Patrick Y. Shinzato1 and Denis F. Wolf1 and Christoph Stiller, “Road Terrain Detection: Avoiding Common Obstacle Detection Assumptions Using Sensor Fusion”, IEEE Intelligent Vehicles Symposium (IV)/2019.

[8] H. G. Jung, Y. H. Lee, H. J. Kang and J. Kim, “Sensor Fusion-Based Lan Detection For LKS+ACC System”, International Journal of Automotive Technology/2019.

[9] Bassam Alrifaee, Matthias Reiter, Jan P. Maschuw, Fr´ederic Christen, Lutz Eckstein, Dirk Abel, “Satellite- and Map-based Long Range Cooperative Adaptive Cruise Control System for Road Vehicles”, International Journal of Digital Application & Contemporary research/2020.

[10] JochemT,LangerD(1996)Fusing radar and vision for detecting, classifying, and avoiding roadway obstacles. In: Proceedings of the IEEE Symposium on Intelligent Vehicles, Tokyo, September, pp 333–338

[11] Vahidi and A. Eskandarian, “Research advances in intelligent collision avoidance and adaptive cruise control,” IEEE Trans/2018.